

**VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky**

# **Technologie pro vývoj moderních webových aplikací RIA**

## **Technologies for Rich Internet Application Development**

## Zadání diplomové práce

Student:

**Bc. Martin Baran**

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Technologie pro vývoj moderních webových aplikací RIA  
Technologies for Rich Internet Application Development

Zásady pro vypracování:

Cílem této diplomové práce je seznámit se s technologiemi vývoje moderních webových aplikací RIA (Rich Internet Applications), na vybraných technologiích vytvořit vhodné ukázky aplikací, a tak porovnat dané technologie z hlediska efektivity programování a výkonu.

- 1) Popište základní principy RIA aplikací.
- 2) Podrobněji popište vybrané technologie pro vývoj těchto aplikací, jejich architekturu a možnosti.
- 3) Navrhněte a vytvořte jednoduchou webovou aplikaci v každé z navržených technologií.
- 4) Na takto vytvořených aplikacích porovnejte technologie z hlediska efektivity a náročnosti programování a z hlediska výkonu.

Seznam doporučené odborné literatury:

Dle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Štěpán Kuchař**

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. Ing. Ivo Vondrák, CSc.  
děkan fakulty

## **Prohlášení**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě, dne 26. 4. 2010

.....  
Bc. Martin Baran

## **Poděkování**

Velmi rád bych poděkoval a vyslovil uznání vedoucímu mé diplomové práce Ing. Štěpánu Kuchařovi za trpělivé vedení a množství praktických rad.

## **Abstrakt**

Cílem této diplomové práce je seznámit se s technologiemi vývoje moderních webových aplikací RIA (Rich Internet Applications), na vybraných technologiích vytvořit vhodné ukázky aplikací a tak porovnat dané technologie z hlediska efektivity programování a výkonu.

## **Klíčové slova**

Rich Internet Application, Adobe Flex, Microsoft Silverlight, Oracle JavaFX, ActionScript, VB.NET, C#, JavaFX Script

## **Abstract**

The objective of this work is to introduce technologies for development of modern web applications RIAs (Rich Internet Applications), to create suitable examples of applications using the selected technologies and to compare the technologies in terms of programming and performance.

## **Keywords**

Rich Internet Application, Adobe Flex, Microsoft Silverlight, Oracle JavaFX, ActionScript, VB.NET, C#, JavaFX Script

## Seznam zkratek a pojmů

<b>AAC</b>	<i>Advanced Audio Coding</i>
<b>AIFF</b>	<i>Audio Interchange File Format</i>
<b>AJAX</b>	<i>Asynchronous JavaScript and XML</i>
<b>API</b>	<i>Application Programming Interface</i>
<b>apod.</b>	<i>a podobně</i>
<b>ASF</b>	<i>Advanced Systems Format</i>
<b>atd.</b>	<i>a tak dále</i>
<b>AVI</b>	<i>Audio Video Interleave</i>
<b>AVM1</b>	<i>ActionScript Virtual Machine verze 1</i>
<b>AVM2</b>	<i>ActionScript Virtual Machine verze 2</i>
<b>B2B</b>	<i>Business to business</i>
<b>B2C</b>	<i>Business to customer</i>
<b>BCL</b>	<i>Base-Class Library</i>
<b>C2D</b>	<i>Core 2 Duo</i>
<b>CGI</b>	<i>Common Gateway Interface</i>
<b>CSS</b>	<i>Cascading Style Sheet</i>
<b>DOM</b>	<i>Document Object Model</i>
<b>EBML</b>	<i>Extensible Binary Meta Language</i>
<b>FLV</b>	<i>Flash Video</i>
<b>GB</b>	<i>Gigabyte</i>
<b>GHz</b>	<i>Gigahertz</i>
<b>GWT</b>	<i>Google Web Toolkit</i>
<b>HD</b>	<i>High-Definition</i>
<b>HTML</b>	<i>HyperText Markup Language</i>
<b>HTTP</b>	<i>HyperText Transfer Protocol</i>
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>IE</b>	<i>Internet Explorer</i>
<b>J2EE</b>	<i>Java 2 Platform, Enterprise Edition</i>
<b>Java SE</b>	<i>Java Platform, Standard Edition</i>
<b>JDK</b>	<i>Java Development Kit</i>
<b>JRE</b>	<i>Java Runtime Environment</i>
<b>JSON</b>	<i>JavaScript Object Notation</i>

<b>JVM</b>	<i>Java Virtual Machine</i>
<b>JWS</b>	<i>Java Web Start</i>
<b>LINQ</b>	<i>Language Integrated Query</i>
<b>MIDI</b>	<i>Musical Instrument Digital Interface</i>
<b>MP2</b>	<i>MPEG layer 2 - Motion Picture Experts Group- layer 2</i>
<b>MP3</b>	<i>MPEG layer 3 - Motion Picture Experts Group layer 3</i>
<b>MPEG</b>	<i>Motion Picture Experts Group</i>
<b>MS</b>	<i>Microsoft</i>
<b>MXML</b>	<i>„Macromedia eXtensible Markup Language“</i>
<b>Obr.</b>	<i>obrázek</i>
<b>OOB</b>	<i>Out Of Browser</i>
<b>OOP</b>	<i>Object-oriented programming</i>
<b>RGBA</b>	<i>Red Green Blue Alpha</i>
<b>RIA</b>	<i>Rich Internet Application</i>
<b>RSS</b>	<i>Really Simple Syndication</i>
<b>SDK</b>	<i>Software Development Kit</i>
<b>SOAP</b>	<i>Simple Object Access Protocol</i>
<b>SP</b>	<i>Service Pack</i>
<b>SSB</b>	<i>Site-Specific Browsers</i>
<b>Tab.</b>	<i>Tabulka</i>
<b>tzn.</b>	<i>to znamená</i>
<b>tzv.</b>	<i>takzvaný</i>
<b>VB</b>	<i>Visual Basic</i>
<b>WAV</b>	<i>Waveform Audio Format</i>
<b>WIN</b>	<i>Windows</i>
<b>WMA</b>	<i>Windows Media Audio</i>
<b>WMV</b>	<i>Windows Media Video</i>
<b>WPF</b>	<i>Windows Presentation Foundation</i>
<b>WPF/E</b>	<i>Windows Presentation Foundation / Everywhere</i>
<b>XAML</b>	<i>Extensible Application Markup Language</i>
<b>XML</b>	<i>Extensible Markup Language</i>

## Obsah

### Seznam zkratk a pojmů

1	Úvod.....	4
2	Rich Internet Application(RIA) .....	5
2.1	Vývoj webových aplikací.....	6
2.2	Desktopové, webové a RIA aplikace .....	6
2.3	Výhody a nevýhody RIA aplikací.....	8
3	Technologie pro tvorbu webových RIA aplikací .....	9
3.1	AJAX .....	10
3.2	Adobe Flash .....	11
3.3	Curl.....	11
3.4	OpenLaszlo .....	12
3.5	Google Gears.....	12
3.6	Mozilla Prism .....	13
4	Adobe Flex .....	15
4.1	Historie.....	15
4.1.1	Macromedia Flex 1.0, 1.5 .....	15
4.1.2	Adobe Flex 2.0 a 2.01 .....	15
4.2	Adobe Flex 3 .....	16
4.2.1	MXML .....	17
4.2.2	ActionScript 3.0 .....	18
4.2.3	Vývojové nástroje .....	19
4.2.4	Běhové prostředí .....	19
4.2.5	Nasazení aplikace do webové stránky.....	20
4.3	Budoucnost.....	20
5	Microsoft Silverlight .....	21
5.1	Historie.....	21
5.1.1	Microsoft Silverlight 1.0 .....	21
5.1.2	Microsoft Silverlight 2.0 .....	21
5.2	Microsoft Silverlight 3.0 .....	21
5.2.1	XAML.....	23



5.2.2	Programovací jazyky.....	23
5.2.3	Vývojové nástroje .....	25
5.2.4	Běhové prostředí .....	25
5.2.5	Nasazení aplikace do webové stránky.....	26
5.3	Budoucnost.....	27
6	Oracle JavaFX.....	28
6.1	Historie.....	28
6.1.1	Sun JavaFX 1.0 .....	28
6.1.2	Sun JavaFX 1.1 .....	28
6.2	Oracle JavaFX 1.2.....	28
6.2.1	JavaFX Script.....	29
6.2.2	Java.....	31
6.2.3	Vývojové prostředí.....	32
6.2.4	Běhové prostředí .....	32
6.2.5	Nasazení aplikace do webové stránky.....	33
6.3	Budoucnost.....	34
7	Porovnání vlastností vybraných technologií .....	35
7.1	Porovnání základních vlastností.....	35
7.2	Animace .....	35
7.3	Velikost výsledného souboru .....	36
7.4	Porovnání možností vývojových prostředí.....	36
7.5	Rozšířenost běhového prostředí .....	38
8	Porovnání z programátorského hlediska .....	39
8.1	Tvorba videopřehrávače.....	39
8.1.1	Podpora video a audio médií.....	40
8.1.2	Komponenty pro přehrávání videí.....	40
8.2	Porovnání technologií z pohledu programování .....	42
9	Výkonnostní testy .....	44
9.1	Test rychlosti načítání a vykreslování grafických prvků .....	44
9.1.1	Způsob měření.....	44
9.1.2	Výsledky testování .....	45
9.2	Test rychlosti zpracování složitějšího algoritmu.....	47
9.2.1	Způsob měření.....	47

9.2.2	Výsledky testování .....	47
10	Závěr .....	49
Seznam použité literatury		
Seznam obrázků		
Seznam tabulek		
Seznam výpisů		
Seznam grafů		
Seznam příloh		

# 1 Úvod

Internet se stále vyvíjí a nabízí nové možnosti využití. Již to nejsou pouze webové stránky a data (dokumenty, média apod.), ale nedílnou součástí se staly také webové (internetové) aplikace. Tyto aplikace uživatelům usnadňují práci, jsou graficky zajímavé a v některých případech dokonce nahrazují programy nainstalované na počítači. V dnešní době již existují webové aplikace, které jsou funkčně podobné textovým editorům, grafickým programům atd. Určitě každý uživatel internetu se s těmito aplikacemi setkal ať již vědomě nebo nevědomě.

Obecně se tyto webové aplikace nazývají RIA. V poslední době se tento pojem často vyskytuje v mnoha odborných článcích, které jsou zaměřené na vývoj internetu. Odborníci nazývají RIA jako další generaci webových aplikací. Osobně si myslím, že i přesto, že se RIA aplikace vyvíjejí již několik let, je stále tento vývoj v počátcích. V budoucnu, pokud se bude s vývojem pokračovat, nám mohou tyto aplikace nabídnout ještě řadu zajímavých možností.

Tato práce se zabývá vysvětlením samotné zkratky RIA a seznámením s technologiemi, které umožňují vývoj těchto aplikací. O tento druh aplikací se zajímám již delší dobu a právě z toho důvodu jsem si zvolil za svou diplomovou práci toto téma. Cílem této práce je porovnání dostupných RIA technologií. Jelikož v současné době existuje mnoho technologií určených pro vytváření RIA aplikací, rozhodl jsem se pro porovnání vybrat tři nejpoužívanější technologie. Těmito jsou podle mého uvážení technologie Adobe Flex, Microsoft Silverlight a Oracle JavaFX. V této práci podrobně popíšu každou z vybraných technologií. Popis bude obsahovat historii, popis architektury, popis aktuální verze, možnosti dostupných vývojových nástrojů a potřebné běhové prostředí. Závěrem práce porovnáám dané technologie z pohledu jejich vlastností a možnosti programování. Pro potřeby porovnání z programátorského hlediska navrhnu a vytvořím aplikaci ve vybraných technologiích. Mou snahou bude naimplementovat strukturálně i funkčně podobné aplikace. Následně vytvořené aplikace porovnáám po stránce výkonu a efektivity programování.

## 2 Rich Internet Application(RIA)

Rich Internet Application je považován za nový směr, kterým se nová generace internetových aplikací ubírá. Tyto aplikace jsou uživatelsky příjemnější a snaží se vzhledem i chováním podobat desktopovým aplikacím. Slovo „Rich“ v termínu má naznačovat, že dané aplikace jsou efektivní, flexibilní, interaktivní a zejména jsou snadno použitelné.

Samotný termín RIA začala v roce 2002 propagovat firma Macromedia. V té době termín označoval webové aplikace, které měly podobné vlastnosti jako desktopové aplikace. Dnes tyto aplikace nemusí být zobrazovány jen pomocí webového prohlížeče, začala také expanze na mobilní zařízení a desktop. Proto se v dnešní době termín RIA považuje spíše za směr než jen za „webové aplikace“. [1]



Obr. 2.1 Oblasti RIA aplikací

Samotné RIA aplikace můžeme rozdělit do tří základních oblastí; podle účelu vytvoření aplikace a podle prostředí, na kterém daná aplikace poběží. Obrázek 2.1 zobrazuje jednotlivé oblasti, jedná se o aplikace webové, desktopové nebo mobilní. [2]

Nejvíce společnosti se zaměřilo zejména na vývoj RIA aplikací určených pro webové prostředí. Také z toho důvodu existuje řada dostupných technologií pro tuto oblast. Tabulka 2.1 ukazuje nejznámější technologie pro každou ze zmiňovaných oblastí.

Webové aplikace		Desktop aplikace	Aplikace pro mobilní zařízení
Adobe Flex	Google Web Toolkit	Adobe AIR	Adobe Flash Lite
Adobe Flash	Microsoft Silverlight	Eclipse RPC	Oracle JavaFX Mobile
AJAX	OpenLaszlo	Microsoft WPF	
Curl	Oracle JavaFX	Mozilla Prism	
Google Gears			

Tab. 2.1 Nejznámější technologie pro tvorbu RIA aplikací v jednotlivých oblastech

Tato práce se zabývá oblastí z pohledu webových aplikací, ostatní oblasti a ani jejich technologie nebudou dále podrobněji popisovány. V dalších částech textu se zkratka RIA bude používat pouze ve významu RIA aplikací určených pro web.

## **2.1 Vývoj webových aplikací**

Na počátku Internetu měly webové stránky pouze textovou podobu ve formě jednoduchých HTML stránek. Postupem času se přidávaly nové elementy do jazyka HTML a webové stránky umožňovaly například zobrazování obrázků. S nástupem kaskádových stylů dostaly stránky určitou formu a také pomocí CSS vlastností byly graficky přívětivější.

Webové stránky měly stále pouze informativní charakter. Postupně se začaly objevovat webové aplikace, které přinesly i funkcionalitu. Tyto aplikace byly postaveny na architektuře klient-server a byly pouze statické. Tedy při každém požadavku uživatele byl poslán HTTP požadavek na daný server, jako odpověď byla poslána stránka ve formě HTML.

Pokrok nastal na počátku 90. let, kdy se začal používat Common Gateway Interface (CGI). Tento protokol umožňuje propojení externí aplikace s webovým serverem a tím poskytuje dynamické zpracování obsahu. Po odeslání HTTP požadavku je možné na straně serveru spustit programy, které zpracují požadavek a pošlou vygenerovanou stránku zpět uživateli.

V roce 1995 představila společnost Sun softwarovou komponentu pod názvem Applet. Aplikace jsou napsané v jazyce Java a je možné je zobrazit přímo v HTML stránce. Pro samotný běh je nutné mít nainstalovány Java plugin a prohlížeč, který s ním dokáže pracovat. Applety mají řadu funkčních omezení, a proto se dnes již často nepoužívají.

Ve stejném roce představila firma Netscape skriptovací jazyk JavaScript, ten umožňuje zpracování některých dynamických prvků pomocí skriptů na straně klienta. I když se z názvu může zdát, že při psaní je využíván jazyk Java, není tomu tak. JavaScript je ovlivněn řadou jazyků a s jazykem Java má podobnou pouze syntaxi. Hlavní výhodou JavaScriptu je přímý zápis do HTML kódu. Server posílá klientovi stránku (i s JavaScriptem), kde se při dané události provede požadovaný kód.

S příchodem DOM byla funkčnost JavaScriptu značně rozšířen. Dnes dokáže pracovat s kompletní stránkou v prohlížeči, specifikovanou jako objektový model. JavaScript se nevyužívá jen pro práci na straně klienta (AJAX), ale lze ho použít také jako „programovací“ jazyk.

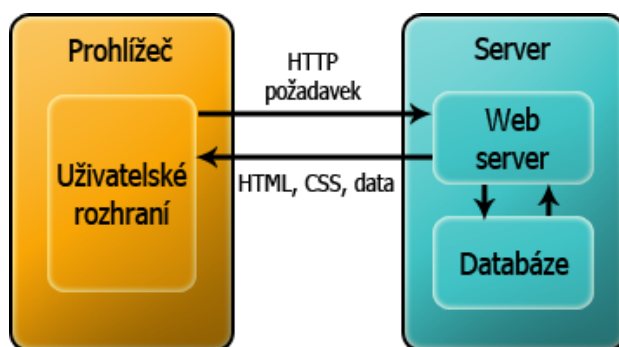
## **2.2 Desktopové, webové a RIA aplikace**

Desktopové aplikace jsou programy, které uživatel spouští na svém počítači, ve většině případů pracují pouze lokálně. Mezi hlavní výhody desktop aplikací patří jejich rychlost, často obsahují bohaté grafické a pro uživatele přívětivé prostředí. Pokud program nepracuje pouze lokálně a je vyžadován přístup k internetu, je často důležité zajistit bezpečné spojení. Další nevýhodou těchto aplikací je v mnoha případech nutná instalace a s tím spojené nároky na hardwarovou část počítače. Také může nastat problém při kompatibilitě s operačním systémem, instalací knihoven apod.

Webové aplikace se snaží část těchto desktopových problémů odstranit. Používá se architektura klient-server, která pracuje na HTTP protokolu – využívá se modelu požadavek/odpověď (anglicky request/response). Na straně uživatele běží tzv. tenký klient

ve formě prohlížeče, který zajišťuje prezentační vrstvu. Uživatel při vyvolání události přes prohlížeč (např. kliknutí na tlačítko) odešle požadavek na server. Server požadavek zpracuje a pošle uživateli zpět odpověď v HTML podobě.

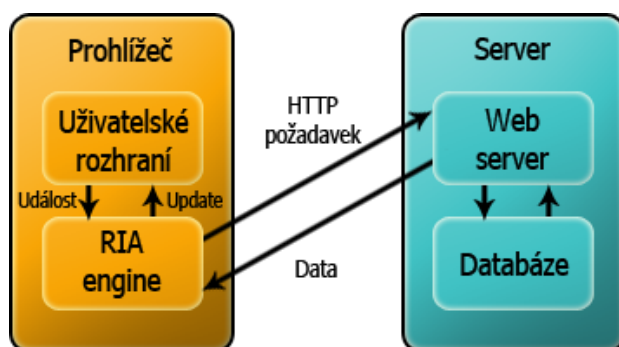
Hlavní výhodou webových aplikací je skutečnost, že nejsou kladeny žádné nároky na hardware. Naopak nutností je mít na daném počítači nainstalován webový prohlížeč, který bude plnit funkci tenkého klienta. V dnešní době jsou již prohlížeče instalovány současně s operačním systémem, a proto není problém přistupovat k webovým aplikacím z jakéhokoli počítače. Tyto aplikace jsou často statické, veškerá logika se provádí na straně serveru a každý požadavek musí projít přes server. Z tohoto důvodu jsou webové aplikace časově náročnější. Vzhled u webových aplikací se provádí pomocí jazyka HTML a kaskádových stylů, které nenabízejí takovou možnost vytváření graficky bohatého prostředí jako u aplikací desktopových.



Obr. 2.2 Model komunikace klasické webové aplikace

RIA aplikace kombinuje výhody desktopových i webových aplikací. Na obrázku 2.3 můžeme vidět obecný model komunikace RIA aplikace, která je postavena na tradičním modelu webových aplikací a je rozšířena o takzvanou klientskou vrstvu. Tato vrstva je umístěna mezi prezenční a logickou vrstvou a umožňuje zpracování požadavku na straně klienta.

RIA aplikace se snaží dosáhnout pro uživatele stejně přívětivého prostředí, které nabízejí desktopové aplikace. Proto řada technologií pro RIA nabízí množství grafických prvků, které byly v minulosti dostupné jen pro desktop aplikace.



Obr. 2.3 Obecný model komunikace RIA aplikací

## 2.3 Výhody a nevýhody RIA aplikací

„Bohaté“ aplikace nabízejí řadu výhod a důvodů na jejich použití. Mezi tyto výhody patří například snadné začlenění aplikace do webové stránky. Na straně klienta stačí pro spuštění pouze prohlížeč a zásuvný modul, který se stará o běh aplikace. Není potřeba žádné údržby, protože aplikace se zcela samy aktualizují při opětovné návštěvě stránky.

RIA nabízí oproti klasickým webovým aplikacím bohatší prostředí (menu, záložky apod.). Podle zvolené technologie umožňuje také mnoho funkcí a možností implementace (funkci drag&drop, výpočty na straně klienta atd.).

Nespornou výhodou je také komunikace se serverem, která probíhá asynchronně. Většina událostí je zpracována na straně klienta a se serverem komunikuje jen při složitých požadavcích nebo při práci s daty. RIA se také snaží optimalizovat přenos dat mezi serverem a tím urychlit samotnou komunikaci.

RIA používá pro svůj běh implementační vrstvu, která je uzavřena vůči okolí. Z toho důvodu ji lze považovat za bezpečnou.

Ačkoliv RIA aplikace nabízejí mnoho výhod, jsou s nimi spojené i určité nevýhody. Mezi ně můžeme zařadit potřebu povolení JavaScriptu na straně klienta, popřípadě instalaci běhového prostředí potřebnou pro spuštění dané aplikace.

Samotné aplikace jsou distribuovány přes internet, je zde tudíž závislost na internetovém připojení. I když se některé RIA technologie snaží tuto nevýhodu odbourat, stále je vyžadováno alespoň příležitostné připojení.

Mezi další nevýhody můžeme rovněž zařadit rychlost stahování a zpracování skriptu na straně klienta. Určitou nevýhodou je také uzavřenost implementační vrstvy vůči okolí, ta může omezovat přístup k lokálním zdrojům. S uzavřeností aplikací je spojena také další nevýhoda, kterou je neviditelnost obsahu aplikace pro internetové vyhledávače.

### 3 Technologie pro tvorbu webových RIA aplikací

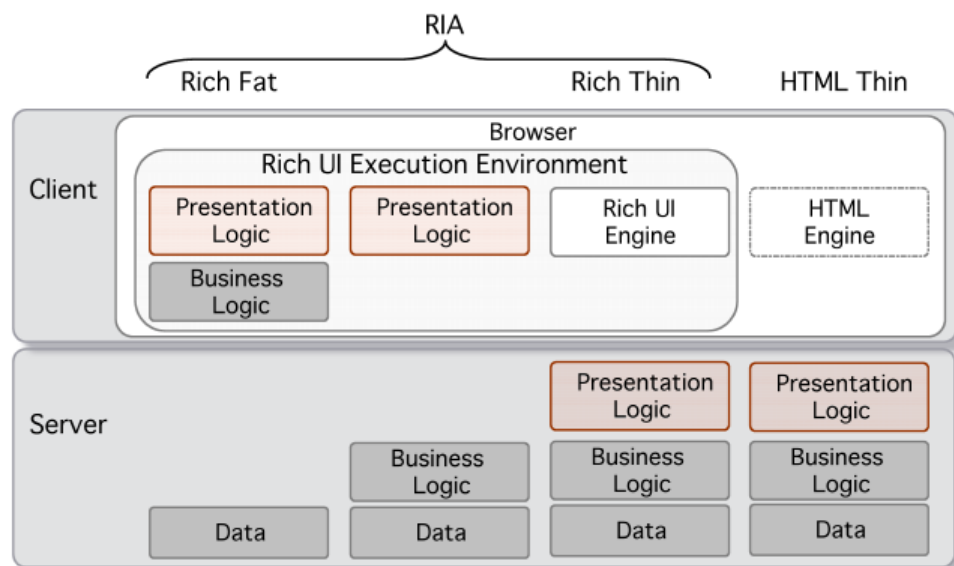
V dnešní době již existuje celá řada technologií, které umožňují tvorbu těchto bohatých aplikací určených pro webové rozhraní. Společně s vývojem těchto aplikací se rozběhl také boj velkých společností o snahu dosáhnout největšího podílu na trhu v oblasti RIA.

RIA aplikace určené pro webové rozhraní můžeme rozdělit do dvou základních skupin podle podpory ve webovém prohlížeči.

První skupinou jsou aplikace, které jsou podporovány přímo ve webovém prohlížeči, tedy nevyžadují instalaci žádného rozšíření. Jsou nezávislé na platformě a využívají uživatelského prostředí prohlížeče. Zde patří zejména technologie AJAX.

Do druhé skupiny RIA aplikací patří technologie, které pro svůj běh na straně klienta vyžadují instalaci běhového prostředí. Příkladem často využívaného prostředí pro běh aplikací může být například Flash Player od firmy Adobe. Běhové prostředí jsou ve formě zásuvných modulů pro webové prohlížeče a samotná instalace je ve většině případů velice jednoduchá a rychlá. Aplikace této skupiny využívá vlastní uživatelské prostředí, z tohoto důvodu také nabízí mnoho grafických možností. Tato skupina je prezentována řadou technologií, z nich nejznámější jsou například Adobe Flash, Adobe Flex, Microsoft Silverlight, JavaFX, CURL a Google Gear.

Na obrázku 3.1 lze vidět architekturu existujících webových aplikací. Napravo je zobrazena klasická HTML aplikace, kde na straně klienta se zobrazuje pouze HTML kód a veškerá logika je zpracována na straně serveru. Ostatní architektury jsou již architektury RIA aplikací, například první dva sloupky zleva představují AJAX a JavaScript. Na obrázku si lze také všimnout, že je snaha přesunout co největší část logiky aplikace ze serveru přímo do webového prohlížeče. Podle toho lze RIA aplikace dále dělit na takzvané tenké a tlusté.



Obr. 3.1 Architektura webových aplikací [6]



### 3.1 AJAX

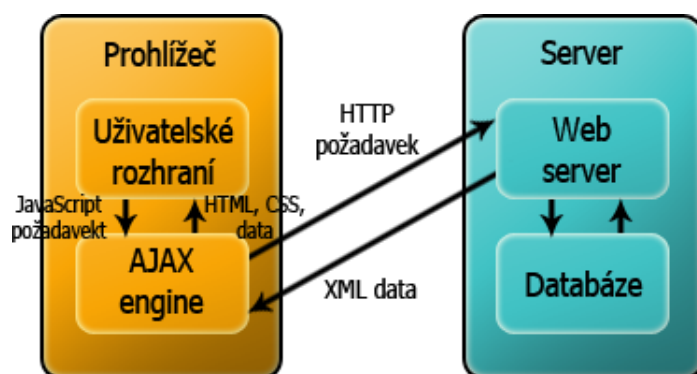
AJAX (Asynchronous JavaScript and XML) není přímo technologie, ale je to označení architektury webových aplikací. Využívá skriptovací jazyk JavaScript a značkovací jazyk XML pro načítání dat ze serveru bez nutnosti znovunačtení stránky. Jak již bylo napsáno, AJAX nepotřebuje ke svému použití žádnou instalaci běhového prostředí.

AJAX využívá pro svůj běh řadu jiných technologií. Pro prezentaci dat je to jazyk HTML a CSS styly. Zobrazování a dynamické změny informací se provádějí pomocí JavaScriptu a modelu DOM. Pro asynchronní výměnu dat se serverem je využíváno rozhraní XMLHttpRequest (podporuje prostý text, XML, HTML, JSON i EBML).

Komunikace se serverem probíhá asynchronně, tzn. nečeká se na odpověď ze strany serveru. Průběh komunikace je názorně ukázán na obrázku 3.2 a probíhá následovně. Při požadavku ze strany klienta se spustí událost JavaScriptu a zavolá se příslušná funkce. Ta vytváří objekt XMLHttpRequest, který pošle požadavek s informacemi o načtené stránce na server. Na straně serveru se požadavek zpracuje a pošle zpět HttpServletResponse ve formátu XML s potřebnými daty. JavaScript analyzuje příchozí XML zprávu a podle ní provede úpravu modelu DOM webové stránky. [7]

AJAX má oproti jiným technologiím značnou výhodu podpory ve webových prohlížečích, nevyžaduje žádnou instalaci zásuvného modulu. Problémy u AJAX aplikací mohou nastat při potřebě používání tlačítek Zpět a Další ve webovém prohlížeči, jelikož nezaznamená historii stránky. Zatím existují pouze techniky, které tento problém řeší pouze částečně (neviditelné IFRAME nebo část adresy za #). Za výhodu a současně i za nevýhodu se považuje samotná asynchronní komunikace. Může nastat problém s rychlostí odezvy při požadavku, kdy uživatel není informován o stavu dění a neví, zda se požadavek zpracovává nebo nebyl odeslán. [8]

Jelikož se AJAX aplikace nevytvářejí jednoduše, představila společnost Google nový nástroj nesoucí název Google Web Toolkit (GWT). Tento framework umožňuje uživateli jednodušší vytváření aplikací pomocí předem vytvořených UI komponent, ty jsou následně zobrazovány jako HTML kód. GTW využívá programovací jazyk Java a při kompilaci je tento kód převeden do podoby JavaScriptu.



Obr. 3.2 Model komunikace AJAX aplikace

### 3.2 Adobe Flash

Flash je vektorový grafický program vydávaný firmou Adobe (dříve Macromedia). Využívá se zejména pro internetovou tvorbu, jako jsou bannery, animace, hry a prezentace.

Používá koncept postavený na časové ose (uživatel si vytvoří klíčové snímky a zvolí přechody mezi nimi). Flash nabízí i možnost vytváření složitějších aplikací a to pomocí implementačního jazyka ActionScript. I přes schopnost vytváření určité funkcionality je tento produkt využíván hlavně grafiky; programátoři používají technologii Adobe Flex.

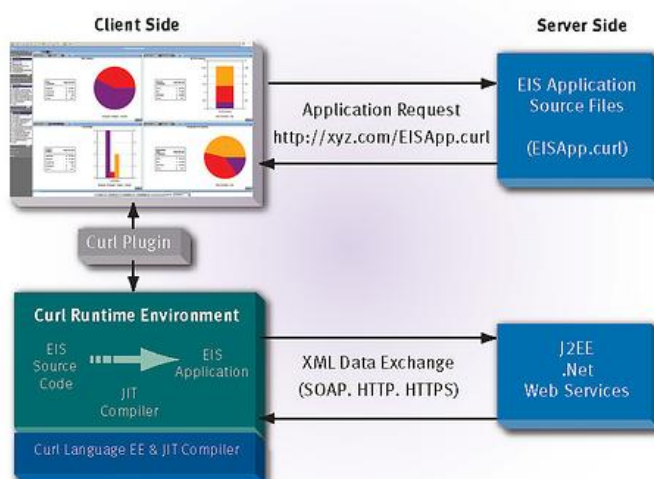
Výstupní soubory jsou ve formátu SWF (případně EXE) a nasazují se do webové stránky jako samostatný objekt. Aby se daný SWF soubor uživateli zobrazil, je potřeba mít nainstalované běhové prostředí Adobe Flash Player ve webovém prohlížeči. Tento doplněk je zdarma k dispozici a v dnešní době je nainstalován asi v 98 % domácnosti. [9]

Hlavním důvodem oblíbenosti Flashe je malá velikost výstupního souboru (příčinou je uchovávání obrázků ve vektorovém formátu) a možnosti tvorby, kterou nabízí. Je multiplatformní, tzn. aplikace bude fungovat ve všech prohlížečích pod každým operačním systémem. Mezi hlavní nevýhody patří možnosti editace. Pokud si uživatel nechoval zdrojové kódy, není možné provést změny v souboru.

### 3.3 Curl

Curl je objektově orientovaný programovací jazyk od společnosti Curl, Inc. & Sumisho Computer Systems Corp. Využívá značkovací jazyk, který je velice podobný jazyku HTML, a skriptovací jazyk, připomínající JavaScript. Nejčastěji se využívá pro interní B2B a B2C aplikace. Curl aplikace jsou kompilovány na straně klienta do nativního kódu. Aby uživatel mohl následně vytvořené applety zobrazit ve svém webovém prohlížeči, je potřeba mít nainstalovaný modul Curl RTE. [10]

Výhodou Curl je podpora OOP programování, které umožňuje definování tříd, procedur apod. Za další výhodu můžeme považovat bezpečnost; aplikace mohou komunikovat pouze se serverem a mají značně omezený přístup k disku uživatele.



Obr. 3.3 Model Curl aplikace [11]

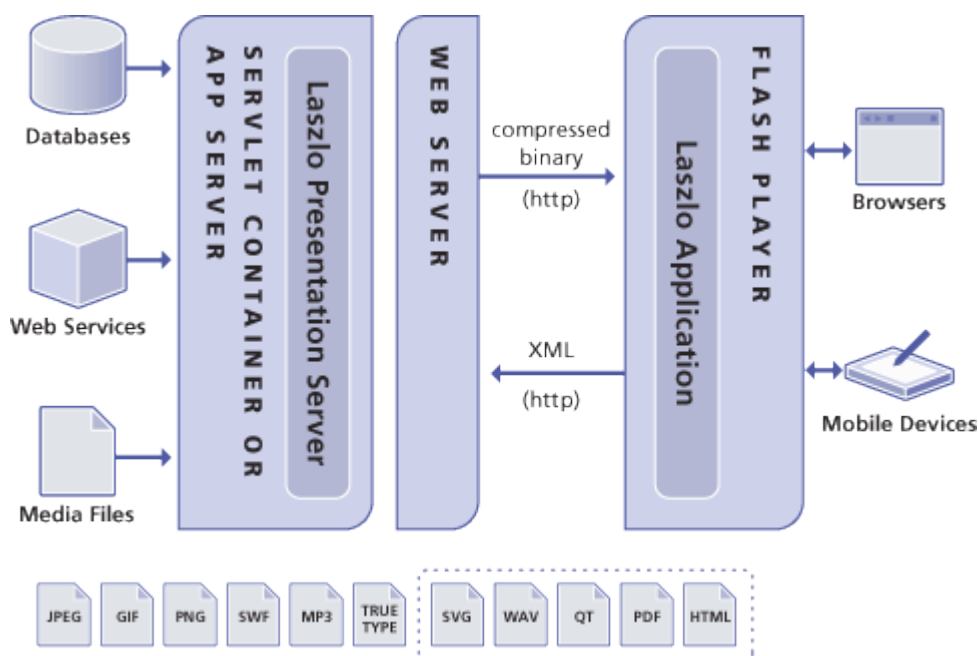
### 3.4 OpenLaszlo

OpenLaszlo je open source (od roku 2004) platforma od firmy Laszlo Systems. Tato platforma se skládá z LZX jazyka a OpenLaszlo Serveru. Jazyk LZX je postaven na XML a JavaScriptu a využívá moderní možností softwarového vývoje.

Pro běh vytvořených aplikací se používá běhové prostředí od firmy Adobe – Adobe Flash Player. Pro spuštění aplikace v přehrávači je nutné mít soubor ve formátu SWF. O to se postará kompilátor obsažený v OpenLaszlo, který vytvoří z JavaScriptu a XML potřebný soubor. [12]

Přístup k datům se provádí pomocí XML souboru poslaného ze serveru (podobný princip jako AJAX). Pomocí OpenLaszlo serveru mohou aplikace také pracovat se SOAP službami. [12]

Hlavní výhodou OpenLaszlo je možnost publikování aplikace pomocí Adobe Flash Player, tím je zajištěna možnost spuštění dané aplikace v každém webovém prohlížeči. Toto usnadnění nese s sebou i menší nevýhodu, jelikož nelze využívat možnosti Flash Playeru API; k dispozici je pouze JavaScript API. Za další nevýhodu můžeme považovat samotné používání jazyka JavaScript, tento skriptovací jazyk nepodporuje v plné míře objektově orientované programování.



Obr. 3.4 Model OpenLaszlo aplikace [3]

### 3.5 Google Gears

V roce 2007 společnost Google začala vyvíjet přídatný plugin pro prohlížeče pod názvem Google Gears. Na vývoji se podílely také společnosti Mozilla, Opera a Adobe.

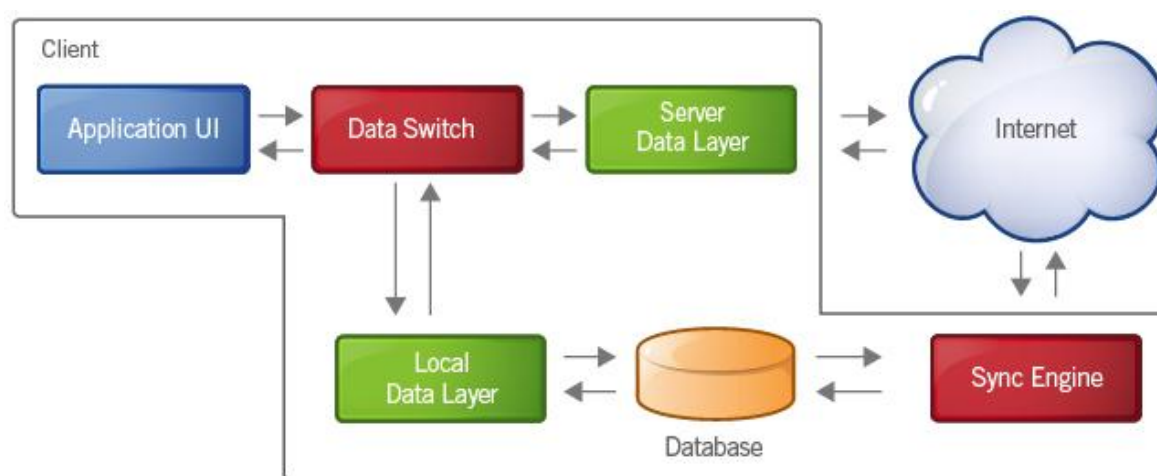
Tento přídatný plugin umožňuje práci v hybridním online/offline módu, přesněji umožňuje cachování webových aplikací a možnost pracovat s těmito aplikacemi offline. Při opětovném

připojení k internetu dojde k synchronizaci. Bohužel nelze takto používat všechny webové aplikace, musí být k tomuto způsobu upravena. [13]

Google přidala podporu offline práce do svých hlavních produktů - Gmail, YouTube, Docs, a další. A díky tomu, že Google vyvíjí tento produkt jako open-source, se pomalu začínají objevovat aplikace využívající Gears také od jiných firem. Příkladem může být podpora offline práce Mail Search od firmy MySpace nebo webová aplikace Writer od společnosti Zoho. [14]

Gears se nesnaží konkurovat AJAXu, Adobe Flexu nebo Microsoft Silverlightu atd. Původně byl vyvíjen pro použití s AJAXem a JavaScriptem a také je za použití JavaScriptu kompatibilní s Flexem i Silverlightem.

Google Gears má podporu v řadě známých prohlížečů jak desktopových (Internet Explorer 6+, Mozilla Firefox 1.5+, Google Chrome, Safari 3.1.1+) tak i v mobilních (IE Mobile 4.1+, Opera Mobile 9.5+). [14]



Obr. 3.5 Návrh architektury aplikace od firmy Google umožňující online/offline práci [15]

### 3.6 Mozilla Prism

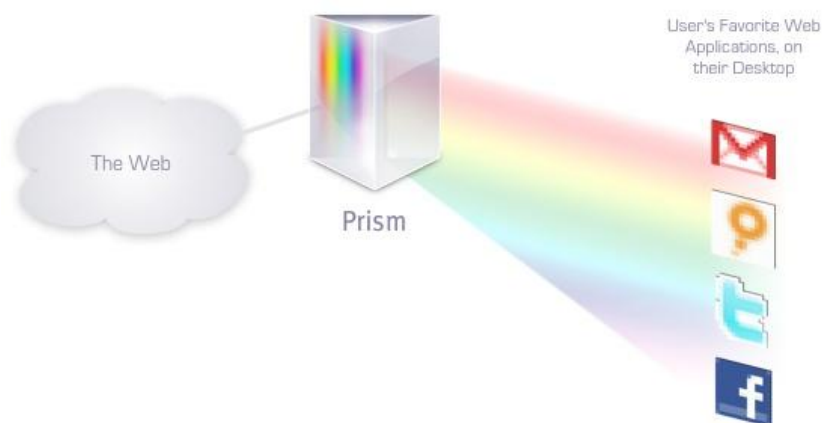
Mozilla Prism (dříve označován jako WebRunner) je jeden z pokusných projektů laboratoře Mozilla Labs. První verze, označována jako verze 0.9, byla vydána v květnu 2008. Poslední vydaný Prism nese označení 1.0 a je stále vedený jako beta verze.

Prism je aplikace, která umožňuje uživatelům přenést jejich oblíbené webové aplikace z prohlížeče na plochu a přímo z plochy tyto aplikace spouštět. Při použití Prism se uživatel vytvoří zástupce na pracovní ploše nebo v nabídce Start. Při spuštění zástupce se daná aplikace otevře v jednoduchém prohlížeči, kde může uživatel dále pracovat s danou webovou aplikací.

Prism je založený na XULRunner, který právě umožňuje běh webové aplikace bez klasického webového prohlížeče. Celý koncept je založen na takzvaném Site-Specific Browsers (SSB). Je to aplikace s vestavěným prohlížečem, který může pracovat pouze s jednou webovou aplikací a neobsahuje žádné nabídky, panely, nástroje jako běžný prohlížeč. [17]

Prism má podobné vlastnosti jako Firefox, proto umožňuje spuštění jakékoliv aplikace, která lze spustit v kompatibilním webovém prohlížeči. Podporuje bohaté internetové technologie, jako HTML, CSS, JavaScript a <canvas>. [18]

Mezi hlavní výhody Prism patří zejména schopnosti autentizace, pamatování velikosti a polohy okna pro každou webovou aplikaci. Pokud jsou v aplikaci použité externí odkazy, stránka se načte do výchozího prohlížeče.



**Obr. 3.6 Mozilla Prism [16]**

## **4 Adobe Flex**

Adobe Flex je sada technologií vyvíjená firmou Adobe Systems. Slouží pro vytváření multiplatformních RIA aplikací.

### **4.1 Historie**

#### ***4.1.1 Macromedia Flex 1.0, 1.5***

V roce 2004 vyšla první verze frameworku pod komerčním jménem Macromedia Flex 1.0. Ve stejném roce byla vydána i verze 1.5. Pro vývoj aplikací byla dostupná sada vývojových nástrojů a knihoven (SDK) nebo přímo vývojové prostředí Flex Builder IDE. Snadná manipulace s daty byla zajištěna pomocí nástroje Flex Data Services, jednalo se o integrovaný nástroj postavený na platformě J2EE. Aplikace byly psány objektově orientovaným jazykem ActionScript ve verzi 2.0. [19]

#### ***4.1.2 Adobe Flex 2.0 a 2.01***

Další verze Flexu vyšla v roce 2006 již pod hlavičkou společnosti Adobe, nesoucí název Adobe Flex 2.0. Firma Adobe zcela změnila licencování a Flex 2 SDK bylo k dispozici bezplatně. SDK obsahovalo knihovnu komponentů a překládač, ovládaný přes příkazový řádek.

Současně společnost Adobe vydala i novou verzi vývojového prostředí Adobe Builder 2, postaveného na platformě vývojového prostředí Eclipse. K dispozici byly dvě verze vývojového prostředí – Standard a Professional. Verze Professional navíc obsahovala knihovnu Flex Charting Componets, ta umožňovala jednoduchou prezentaci dat pomocí grafů. [19]

Obdobně jako v první verzi Flex se pro práci s daty využíval Flex Data Services. Novinkou byla synchronizace dat a tzv. conflict management sloužící pro řízení konfliktů.

Nová verze Adobe Flex 2 přinesla také změnu programovacího jazyka. Začal se používat objektově orientovaný jazyk ActionScript ve verzi 3.0. Ten je již plně v souladu se standardem ECMAScript a umožňuje definovat logiku aplikace na straně klienta.

Pro běh samotných Flexových aplikací se začalo nově používat běhové prostředí Adobe Flash Player 9.

Rozšířenou verzi Adobe Flex 2.01 vydala společnost Adobe v roce 2007. Tato menší aktualizace přinesla možnost přidávání modulů a testování vytvářených aplikací. Novinkou byla také podpora CSS stylů.



Obr. 4.1 Sada nástrojů pro Adobe Flex 2.0 [20]

## 4.2 Adobe Flex 3

Adobe Flex 3 je bezplatný open source framework určený pro vytváření interaktivních aplikací, které lze spustit shodně ve všech významných operačních systémech a prohlížečích. Byl vydán v roce 2008 společností Adobe, poslední rozšíření této verze bylo vydáno v březnu 2009. Vytvořené aplikace pomocí Flexu využívají pro svůj běh ve webovém prohlížeči běhové prostředí Adobe Flash Player 9 (a vyšší). Lze je spustit samostatně přímo na pracovní ploše pomocí runtime prostředí Adobe AIR.

Adobe Flex pro vytváření aplikací používá dva jazyky. První z nich je jazyk MXML, který slouží pro definování vzhledu, rozvržení a chování aplikace. Logika na straně klienta se definuje pomocí objektově orientovaného jazyka ActionScript 3.0. Při následné kompilaci je vytvořen soubor SWF, jenž je spustitelný pomocí již zmiňovaného Adobe Flash Playeru. [21]

Framework Flex obsahuje také předem definovanou knihovnu tříd a aplikačních služeb. Do těchto služeb patří navazování dat, řízení přetížení, systém stylů apod. Dále framework obsahuje knihovnu komponent, která nabízí velké množství ovládacích prvků uživatelského rozhraní (od jednoduchých tlačítek po komplexní datové mřížky). Lze také využívat rozsáhlou knihovnu „flash“, která nabízí řadu možností.

Hlavní novinkou v nové verzi je integrace s Adobe AIR. Mimo již zmiňované možnosti spouštět webové aplikace přímo na pracovní ploše, umožňuje toto běhové prostředí také běh aplikací v takzvaném online/offline módu. [21]



Obr. 4.2 Framework Flex 3

#### 4.2.1 MXML

MXML je značkovací jazyk, založený na jazyku XML, sloužící pro popis uživatelského rozhraní. Se samotnou zkratkou MXML přišla již firma Macromedia v roce 2004. Oficiálně tato zkratka nemá žádný význam, ale řada vývojářů ji překládá jako „Macromedia eXtensible Markup Language“. [22]

MXML se zejména používá pro definování rozložení a vzhledu aplikace, ale lze v něm rovněž zapisovat logiku a chování dané aplikace, přístup ke zdrojům dat na serveru, popřípadě vazby mezi komponenty apod.

Částečně se MXML podobá jazyku HTML, také obsahuje značky pro definici uživatelského rozhraní. Práce s ním je obdobná jako s HTML, ovšem MXML je strukturovanější a obsahuje větší množství značek. Mezi vizuální prvky, které můžeme do aplikace přidat, patří například tlačítka, textové pole, nabídky, tabulky atd. Lze vložit rovněž takzvané nevizuální komponenty jako například webové služby, datové vazby, efekty a animace. Tento značkovací jazyk umožňuje rovněž vytváření vlastních komponent.

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
</mx:Application>
  
```

Výpis 4.1 Základní deklarace MXML

Základní deklarace MXML je zobrazena ve výpisu 4.1. První řádek je povinný pro každý XML dokument a informuje o použité verzi a kódování. V České republice se nejčastěji používá kódování UTF-8, ale je možno používat i jiné. Na druhém řádku je kořenová značka „mx:Application“, zde se pomocí atributů zadávají základní informace o aplikaci např. jmenné prostory, případně lze definovat rozvržení, výšku a šířku aplikace. Do této kořenové značky se také vkládají všechny ostatní vizuální prvky.

Vzhled Flex aplikace lze také vytvářet pomocí jazyka ActionScript. Ve výpisu 4.2 a 4.3 je znázorněno vytvoření jednoduchého tlačítka v jazyce MXML a ActionScript. Můžeme vidět, že vytvoření tlačítka pomocí jazyka ActionScript je značně složitější a náročnější.



```
<mx:Button id="Button1" label="Tlačítko" click="Akce()"/>
```

Výpis 4.2 Vytvoření tlačítka v jazyce MXML

```
import mx.controls.Button;  
  
var Button1:Button = new Button();  
Button1.label = "Tlačítko";  
Button1.addEventListener("click", Akce());
```

Výpis 4.3 Vytvoření tlačítka v jazyce ActionScript 3.0

### 4.2.2 ActionScript 3.0

ActionScript je objektově orientovaný skriptovací jazyk založený na ECMAScript. Byl také vytvořen firmou Macromedia a používal se zejména pro aplikace vyvíjené pomocí Micromedia Flash. Tento jazyk nabízí programátorům vytvářet komplexní internetové aplikace.

ActionScript ve verzi 3.0 byl vydán společně s Adobe Flex 2 v roce 2006. Verze 3.0 je architektonicky i koncepčně odlišná a přinesla řadu změn. Došlo ke zlepšení zpracování XML, výrazně se změnilo objektové API, nově jazyk umožňuje vytvářet pokročilé konstrukce tříd. Za zmínku také stojí přidání nových primitivních objektů a nového Event-frameworku. ActionScript, stejně jako ostatní objektově orientované jazyky, umožňuje definovat třídy, vytvářet vlastnosti a metody, využívat dědičnost a další možnosti OOP pro snadnější vytváření strukturovaného kódu. [24]

Kód ActionScript můžeme zapisovat třemi způsoby. První možnost je zapisovat kód přímo do daného elementu pomocí složených závorek a tečkové notace; takto lze zapisovat jen velmi jednoduché příkazy. Další způsob zápisu kódu je přímo do souboru MXML s použitím vyhrazené značky „Script“ (použito v ukázkové aplikaci Hello World). Poslední možnost je vytvoření samostatného souboru nesoucí příponu .as, do kterého je umístěn kód ActionScriptu. Následně je nutné tento soubor připojit k MXML souboru pomocí elementu „Script“ a jeho atributu „source“.

```
<?xml version="1.0" encoding="utf-8"?>  
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" width="150"  
height="100">  
  <mx:Script>  
    <![CDATA[  
      public function HelloWorld():void  
      {  
        Label.text = "Hello World!";  
      }  
    ]]>  
  </mx:Script>  
  <mx:VBox horizontalAlign="center" verticalAlign="middle" width="150" height="100">  
    <mx:Label id="Label" text=""/>  
    <mx:Button label="Click Me" click="HelloWorld()"/>  
  </mx:VBox>  
</mx:Application>
```

Výpis 4.4 Ukázková Flex aplikace Hello World

### 4.2.3 Vývojové nástroje

Flex nabízí dvě možnosti vývoje RIA aplikací. Jeden ze způsobů zahrnuje použití Flex SDK, který je zdarma k dispozici. Aplikace se napíše v libovolném textovém editoru v MXML nebo ActionScript jazyce. Je důležité soubor uložit s koncovkou .mxml. Pomocí kompilátoru obsaženého v SDK se následně aplikace zkompiluje do souboru SWF - k tomu je vyhrazen příkaz „mxmhc cesta/nazev\_aplikace.mxml“.

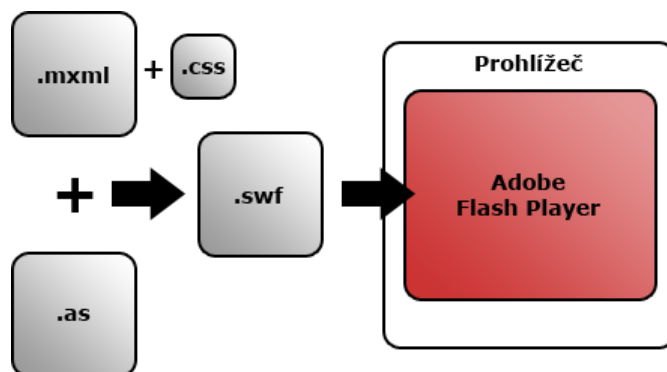
Druhou možností je použití vývojového prostředí od firmy Adobe s názvem Adobe Flex Builder 3. Tento vývojářský nástroj je zcela založen na open source prostředí Eclipse. Využití nástroje umožňuje snadněji a rychleji vyvíjet Flex aplikace. Adobe Flex Builder 3 obsahuje kompletní framework, který zahrnuje vlastní kompilátor, knihovnu komponent a debugger. Dále nabízí vizuální návrh uživatelského rozhraní bez nutnosti psát MXML kód. Umožňuje inteligentní programování, automatické doplňování kódu, zobrazování struktury aplikace, krokování při ladění a v neposlední řadě snadné zveřejnění aplikace. Tento nástroj je ovšem placený a vydává se ve dvou verzích - Standard a Premium. Premium verze navíc nabízí nástroje pro ladění aplikací. Jedná se o různé profilovací nástroje určené pro sledování paměti a výkonu, testování výkonu vizualizace dat apod. Tato verze také nabízí možnost vytváření takzvaného automatického testování. [25]

### 4.2.4 Běhové prostředí

Webové aplikace vytvořené pomocí technologie Flex potřebují pro svůj běh přehrávač Adobe Flash Player. Tento přehrávač je k dispozici pro všechny známější prohlížeče. Jelikož není přímo součástí webových prohlížečů, jsou uživatelé nuceni si ho doinstalovat ve formě zásuvného modulu. Podle průzkumu firmy Adobe je tento přehrávač nainstalován na více než 98 % počítačů.

Pro běh aplikace vytvořené ve frameworku Flex 3 je potřeba Adobe Flash Playeru ve verzi 9.0 nebo vyšší. Obsahuje totiž ActionScript Virtual Machine (AVM2), který zpracovává kód ActionScriptu 3. Pro kompatibilitu s aplikacemi napsanými v předchozích verzích ActionScriptu, obsahuje zároveň první verzi AVM. [19]

Desktopové aplikace jsou zpracovávány pomocí Adobe AIR. Jedná se o modul pro různé operační systémy, který umožňuje využívat webové technologie přímo z desktopu. Při použití Adobe AIR lze přistupovat také k lokálním souborům a systémovým zdrojům na daném počítači.



Obr. 4.3 Struktura Flex aplikace

### 4.2.5 Nasazení aplikace do webové stránky

Pro zobrazení aplikace na webové stránce se vkládá kód podobný výpisu 4.5. Do HTML značky „noscript“ je vložen objekt, ve kterém jsou základní informace potřebné pro spuštění v Adobe Flash Playeru. Samotná stránka, do které se vkládá Flex aplikace, musí obsahovat vygenerovaný kód v jazyce JavaScript. Tento kód zjišťuje, zda je na daném zařízení nainstalovaný Flash Player; následně se také stará o spuštění tohoto přehrávače.

```
<noscript>
  <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    id="HelloWorld" width="150" height="100"
    codebase="http://fpdownload.macromedia.com/get/flashplayer/current/swflash.cab">
    <param name="movie" value="HelloWorld.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#869ca7" />
    <param name="allowScriptAccess" value="sameDomain" />
    <embed src="HelloWorld.swf" quality="high" bgcolor="#869ca7"
      width="150" height="100" name="HelloWorld" align="middle"
      play="true"
      loop="false"
      quality="high"
      allowScriptAccess="sameDomain"
      type="application/x-shockwave-flash"
      pluginspage="http://www.adobe.com/go/getflashplayer">
    </embed>
  </object>
</noscript>
```

Výpis 4.5 HTML kód Flex aplikace vkládaný do webové stránky (ukázka Hello World)

## 4.3 Budoucnost

Nástupcem Adobe Flex 3 je Adobe Flex 4 s kódovým označením Gumbo. Firma Adobe již vydala dvě beta verze - první byla zveřejněna v červnu a druhá v září 2009. Vydání finální verze se očekává v první polovině roku 2010.

Hlavní novinkou v nové verzi, kterou firma Adobe již oznámila, je komponentová architektura „Spark“ - ta odděluje data a chování komponenty. Kompilace v nové verzi má být 2x - 7x rychlejší než ve verzi 3. Pro práci s proměnnými nově přibude možnost 2-way data binding, která bude umožňovat obousměrnou komunikaci mezi proměnnými. Adobe Flex 4 by měl obsahovat bohatší CSS selektory. [26]

V oblasti vzhledu aplikace bude v nové verzi možné vytvářet primitivní grafiku (čáry, čtverce, obdélníky apod.) přímo v souboru MXML (u staré verze Flex 3 je nutné volat grafické API Flash Player z ActionScriptu nebo použít knihovny třetích stran). Nová verze také nabízí podporu 3D efektu, nové možnosti animací apod. Pro lepší spolupráci mezi designérem a programátorem byl vytvořen nový formát FXG, který má umožňovat lepší přenositelnost grafiky mezi programy firmy Adobe. [26]

## **5 Microsoft Silverlight**

Microsoft Silverlight je webový framework vytvořený firmou Microsoft. Umožňuje vytváření interaktivních webových aplikací podobně jako Adobe Flex. Tento framework je postaven na technologii Windows Presentation Foundation (WPF) a .NET Framework.

### **5.1 Historie**

#### **5.1.1 Microsoft Silverlight 1.0**

V září 2007 představila společnost Microsoft první verzi Silverlight, s kódovým označením Windows Presentation Foundation / Everywhere (WPF/E). Tato nástavba byla podporována pouze prohlížeči firmy Microsoft – Internet Explorer.

Návrh aplikace se prováděl pomocí značkovacího jazyka XAML. Samotná funkcionality byla vytvářena pomocí skriptovacího jazyka JavaScript. Práce s daty byla omezena pouze na formát XML a JSON. [28]

Silverlight 1.0 dokázal zpracovávat uživatelské vstupy z klávesnice, z myši apod. Zvládal nejen vykreslování bitmapových obrázků a tvorbu animací, ale také složitější práci s vektorovou grafikou. Umožňoval také přehrávání audio a video multimédií – přesněji formátů MP3, WMA, WMV7, WMV8 a WMV9/VC-1. [28]

#### **5.1.2 Microsoft Silverlight 2.0**

Nová verze Silverlight 2.0 byla uvedena v říjnu 2008. Původní označení bylo Silverlight 1.1, z důvodu značných změn firma Microsoft rozhodla verzi povýšit na verzi 2.0.

Hlavní změnou byl přechod na .NET Framework 3.0. Ten umožňoval psát logiku aplikace v jakémkoliv .NET jazyku – C# nebo VB.NET. Později přibyla také podpora jazyků IronPython, IronRuby a F#. [28]

.NET framework rozšířil Silverlight o část WPF grafických prvků, tím vznikla podpora pro tvary, dokumenty, media apod. Přibýlo také více než třicet nových ovládacích prvků pro uživatelské rozhraní (TextBox, CheckBox atd.) a prvků pro práci s daty (DataGrid, ListBox). [28]

Do Silverlight 2.0 byla přidána knihovna Base-Class Library (BCL), která obsahuje třídy pro práci s kolekcemi, regulárními výrazy, přístupy k datům apod. Také obsahuje podporu dotazovacího jazyka LINQ pro práci s daty. Od předchozí verze, kde byla možnost zpracovávat data jen ve formátu XML a JSON, umožňuje nově pracovat s RSS. Pro jednodušší práci s XML soubory byly také přidány třídy XMLReader a XMLWriter. [28]

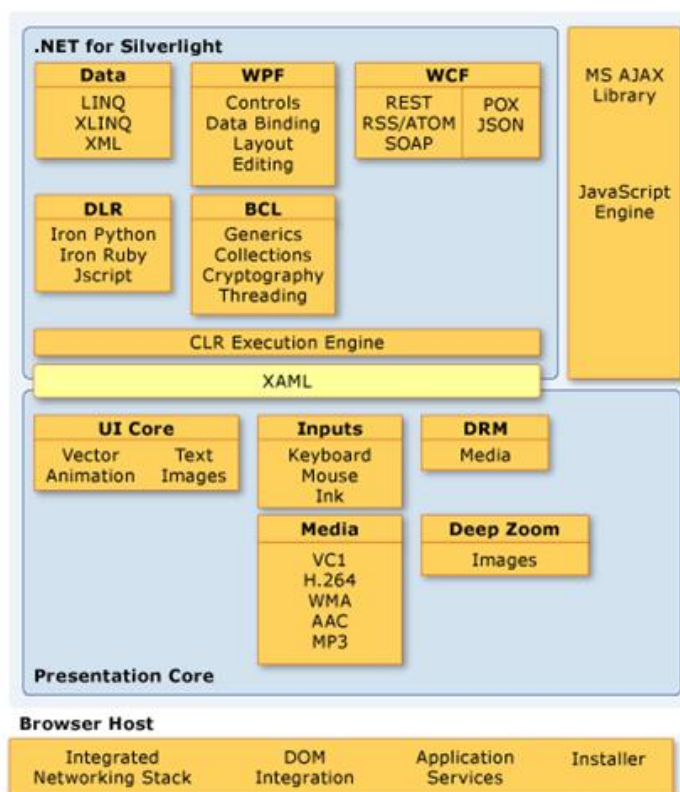
### **5.2 Microsoft Silverlight 3.0**

Silverlight ve verzi 3.0 byl prvně představen v září 2008 na IBC 2008 v Amsterdamu, měsíc po uvedení Silverlight 2 na trh. Při příležitosti odhalení produktu na MIX08 v Las Vegas v březnu 2009 vyšla současně i první beta verze. Finální verzi Microsoft uvolnil již v červenci 2009. [28]

Architektura platformy Silverlightu je rozdělena na dvě části. První část tvoří prezenční vrstva, ta obsahuje komponenty a služby na generování uživatelského rozhraní a interakci s ním. Do uživatelského rozhraní spadá práce s textovým výstupem, animacemi, referováním vektoru a bitmap atd. Za interakci je považovaná jakákoliv událost vytvořena uživatelem – vstup z klávesnice, události pomocí myši apod. Druhou část architektury tvoří instalační a aktualizací komponenty pro webové prohlížeče. [29]

Silverlight pro definování vzhledu aplikací používá deklarativní jazyk XAML. Ten umožňuje vytvářet animace a vazby na data. Pro psaní logické části aplikace lze použít jakýkoliv jazyk z .NET Frameworku (C#, VB.NET). .NET Framework obsahuje velké množství rozsáhlých knihoven, které lze při vytváření aplikací používat. Celá Silverlight aplikace je zkomprimovaná do souboru XAP (ZIP formát). Ke spuštění tohoto formátu v prohlížeči je potřeba mít nainstalované runtime prostředí Silverlight Plug-in. Běh aplikace přímo na klientském počítači zajišťuje OOB (Out Of Browser). [29]

Mezi novinky, které tato verze přináší, patří právě možnost běhu aplikace přímo na klientském počítači. Přibyly také nové komponenty, převážně komponenty ze Silverlight Toolkitu. V oblasti médií došlo k navýšení rychlosti při přehrávání videa a možnosti plynulého přehrávání videa v HD kvalitě. Silverlight také nově podporuje 3D pohled (3D transformace s 2D objekty), alpha blending atd., vše pomocí hardwarové akcelerace. Novinkou je také LocalConnection API, který umožňuje komunikaci mezi více spuštěnými Silverlight aplikacemi na jednom počítači bez ohledu na webový prohlížeč. [30]



Obr. 5.1 Architektura Microsoft Silverlight 3.0 [29]

### 5.2.1 XAML

XAML (Extensible Application Markup Language) je deklarativní jazyk založený na XML a je vyvíjený společností Microsoft. Původně zkratka znamenala Extensible Avalon Markup Language, kde Avalon bylo kódové označení pro WPF. [31]

XAML se používá pro definování prvků uživatelského rozhraní, vazby mezi daty, událostmi apod. Při použití vizuálního návrhu lze vzhled aplikace vytvářet velice snadno, XAML kód je současně generován.

```
<UserControl
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  x:Class="SilverlightApplication.MainPage"
  Width="640" Height="480">
  <Grid x:Name="LayoutRoot" Background="White"/>
</UserControl>
```

Výpis 5.1 Základní deklarace XAML

Základní struktura XAML dokumentu je velice jednoduchá, je zobrazena ve výpisu 5.1. Klíčovou značkou je „UserControl“, v jeho atributech jsou vypsané používané jmenné prostory a název výchozí třídy (atribut „Class“). Lze zde přímo nastavit také základní vlastnosti aplikace, například výšku a šířku. Ostatní grafické prvky se mohou vkládat přímo do „UserControl“, častěji se však využívají layoutové prvky pro snadnější návrh vzhledu aplikace.

Stejně jako u technologie Adobe Flex je možno vzhled aplikace popsat pomocí programovacího jazyka. Přesněji za použití jazyka C# nebo VB.NET.

```
<Button x:Name="Button1" Content="Tlačítko" Click="Akce"/>
```

Výpis 5.2 Vytvoření tlačítka v jazyce XAML

```
Button Button1 = new Button();
Button1.Content = "Tlačítko";
Button1.Click += new System.Windows.RoutedEventHandler(Akce);
```

Výpis 5.3 Vytvoření tlačítka v jazyce C#

```
Dim Button1 As New Button()
Button1.Content = "Tlačítko"
AddHandler Button1.Click, AddressOf Akce
```

Výpis 5.4 Vytvoření tlačítka v jazyce Visual Basic .NET

### 5.2.2 Programovací jazyky

Funkcionalitu Silverlight aplikací lze psát pomocí programovacího jazyka C# verze 3.0 nebo Visual Basic .NET (VB.NET). Oba jazyky vyšly v roce 2007 společně s .NET Frameworkem

3.5 a nabízejí stejné programovací možnosti. C# i VB.NET jsou objektově orientované jazyky, lze tedy při vytváření aplikací například používat třídy, dědičnost, rozhraní a generické datové typy. [32]

Programovací jazyk Basic vznikl v 60. letech minulého století, jeho účel byl zaměřený na výuku. Samotný Visual Basic přišel až s nástupem Windows; vývojové prostředí umožňovalo jednoduše vytvářet uživatelské rozhraní. Visual Basic (VB.NET), jak ho známe v současnosti, vznikla společnost Microsoft s nástupem platformy .NET. Jazyk se stal kompletně objektově orientovaný, přibyla také řada knihovnicí funkcí z .NETu. Základní myšlenka jazyka byla ovšem zachována a tento jazyk se považuje za snadno naučitelný. Syntaxe VB.NETu kvůli srozumitelnosti obsahuje mnoho klíčových slov, které jsou pro začátečníky lehce pochopitelné. Příkazy se neoddělují pomocí středníků, jako je to u mnoha jiných jazyků, ale pomocí konce řádku. Tento jazyk je také case-insensitive, tedy nezáleží na velikosti písmen v klíčových slovech, názvech proměnných, metodách apod. [32]

Historie programovacího jazyka C# je značně kratší než u VB.NET, byl vyvíjen firmou Microsoft společně s .NET Frameworkem. Jazyk se stal velmi oblíbený a díky syntaxi, která je velmi podobná jazykům C/C++ a Java, je přechod pro vývojáře značně jednodušší. C# patří mezi case-sensitive jazyky, tudíž při psaní kódu záleží vždy na velikosti písmen. [32]

```
<UserControl
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:dataInput="clr-
namespace:System.Windows.Controls;assembly=System.Windows.Controls.Data.Input"
  xmlns:controls="clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  x:Class="HelloWorld.MainPage"
  Width="150" Height="100" mc:Ignorable="d">

  <Grid x:Name="LayoutRoot" Background="White">
    <dataInput:Label x:Name="Label" Margin="41,26,41,0" Content="" VerticalAlignment="Top"
HorizontalAlignment="Center"/>
    <Button Margin="0,46,0,32" Content="Click Me" d:LayoutOverrides="Height"
Click="Button_Click" HorizontalAlignment="Center"/>
  </Grid>
</UserControl>
```

Výpis 5.5 Ukázková Silverlight aplikace Hello World (vzhled aplikace - XAML)

```
using System;
using System.Windows;
using System.Windows.Controls;

namespace HelloWorld
{
  public partial class MainPage : UserControl
  {
    public MainPage()
    {
    }
  }
}
```

```

    {
        // Required to initialize variables
        InitializeComponent();
    }

    private void Button_Click(object sender, System.Windows.RoutedEventArgs e)
    {
        Label.Content = "Hello World!";
    }
}

```

Výpis 5.6 Ukázková Silverlight aplikace Hello World (funkcionalita aplikace - C#)

```

Partial Public Class MainPage
    Inherits UserControl

    Public Sub New()
        InitializeComponent()
    End Sub

    Private Sub Button_Click(ByVal sender as Object, ByVal e as System.Windows.RoutedEventArgs)
        Label.Content = "Hello World!"
    End Sub
End Class

```

Výpis 5.7 Silverlight aplikace Hello World (funkcionalita aplikace - VB.NET)

### 5.2.3 Vývojové nástroje

Pro vývoj Silverlight aplikací Microsoft nabízí celkem tři produkty: Microsoft Expression Blend, Microsoft Visual Studio 2008 SP1 a Microsoft Visual Web Developer 2008 SP1. Expression Blend je doporučen pro tvorbu vzhledu Silverlight aplikace. Nabízí vizuální návrh pro rozmístění, manipulaci a modifikaci všech prvků. V tomto programu lze také psát funkcionalitu, avšak pro psaní logiky aplikace je doporučeno používat nástroje Visual Studio 2008 nebo Visual Web Developer 2008. V těchto vývojových nástrojích lze provádět testování a ladění dané aplikace (ve Web Developer 2008 jsou tyto funkce částečně omezené). Z pohledu vytváření vzhledu aplikací obsahují jen základní náhled na vzhled aplikace a přidávání grafického prvku lze učinit přetáhnutím z nabídky ovládacích prvků. Pro vytváření Silverlight aplikací je nutné pro oba tyto programy nainstalovat doplněk Microsoft Silverlight 3 Tools for Visual Studio 2008 SP1. [33]

Aplikace Silverlightu 2 lze vyvíjet i v open-source programu Eclipse, nutno doinstalovat příslušný plugin. V létě 2010 je naplánováno vydání doplňku s podporou Silverlight 3.0. [34]

### 5.2.4 Běhové prostředí

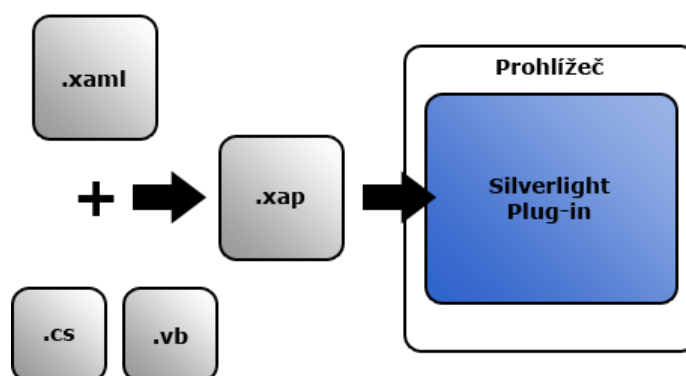
Silverlight aplikace mohou běžet ve webovém prohlížeči nebo přímo z plochy počítače. Běh aplikací ve webovém prohlížeči zajišťuje Silverlight Plug-in ve verzi 3, ten je potřeba



nainstalovat (je dostupný pro prohlížeče Intel Explorer, Mozilla Firefox a Safari). Tento modul je postaven na technologii WPF a .NET Framework, není tedy potřeba instalace .NET Frameworku na klientském počítači.

Spouštění Silverlight aplikací na operačním systému Linux umožňuje běhové prostředí Moonlight. Je to bezplatná a open-source implementace Silverlight frameworku, která je vyvíjena pod vedením společnosti Novell. Poslední verze Moonlight 2.0 (Silverlight 2) byla vydána v prosinci 2009. Nová verze Moonlight 3 s podporou Silverlight 3 je očekávána počátkem roku 2010.

Běh aplikací mimo webový prohlížeč zajišťuje novinka této verze – Out-of-Browser, ten umožňuje nainstalovat Silverlight aplikaci do systému. Pokud je současně aplikace navržena pro lokální instalaci, lze s ní pracovat přímo v offline režimu.



Obr. 5.2 Struktura Silverlight aplikace

### 5.2.5 Nasazení aplikace do webové stránky

Silverlight aplikace je zapouzdřena do HTML značky „object“. V parametrech lze najít název spouštěného souboru, minimální verzi Silverlight Plug-inu potřebnou ke spuštění aplikace atd. Všechny chybové hlášky jsou zpracovány v JavaScript metodě „onSilverlightError“, která se musí vyskytovat na dané stránce.

```

<form id="form1" runat="server" style="height:100%">
  <div id="silverlightControlHost">
    <object data="data:application/x-silverlight-2," type="application/x-silverlight-2" width="100%"
height="100%">
      <param name="source" value="HelloWorld.xap"/>
      <param name="onError" value="onSilverlightError" />
      <param name="background" value="white" />
      <param name="minRuntimeVersion" value="3.0.40624.0" />
      <param name="autoUpgrade" value="true" />
      <a href="http://go.microsoft.com/fwlink/?LinkId=149156&v=3.0.40624.0" style="text-
decoration:none">  </a>
    </object><iframe id="_sl_historyFrame"

```

```
style="visibility:hidden;height:0px;width:0px;border:0px"></iframe></div>
</form>
```

Výpis 5.8 HTML kód Silverlight aplikace vkládaný do webové stránky (ukázka Hello World)

### 5.3 Budoucnost

Společnost Microsoft nezahálí ve vývoji Silverlightu a již v listopadu 2009 vydala první beta verzi Silverlight 4, čtyři měsíce po oficiálním uvedení Silverlight 3. Datum vydání konečné verze ještě není stanoven, očekává se v první polovině roku 2010. [35]

Mezi novinky, které tato verze obsahuje, patří šedesát nových ovládacích prvků pro snadnější vytváření interaktivních aplikací (např. prvek RichTextBox), možnost používání webových kamer a mikrofonu. Nová verze také nabídne možnost běhu aplikace mimo prohlížeč a to bez nutnosti úpravy kódu. [35]

## 6 Oracle JavaFX

JavaFX je softwarová platforma vyvinuta firmou Sun Microsystems, v dnešní době je tato technologie ve vlastnictví firmy Oracle. Je postavena na bázi platformy Java a umožňuje vytvářet stejně jako její konkurence bohaté aplikace.

### 6.1 Historie

#### 6.1.1 Sun JavaFX 1.0

Společnost Sun Microsystems v prosinci 2008 představila nejnovější inovaci platformy Java, byl to produkt s označením JavaFX. Tato platforma měla sjednotit technologii Java na všech zařízeních. Aplikace se programují v novém skriptovacím jazyce JavaFX Script, ten umožňuje stejný model vývoje pro různá nasazení.

Platforma obsahovala tři klíčové komponenty (JavaFX Development Environment, JavaFX Production Suite a JavaFX Desktop) a podporu pro video. [36]

JavaFX Development Environment obsahovala kompilátor a nástroje pro práci s knihovnami (grafika, média, webové služby atd.). Pro pohodlnější vývoj aplikací byl vydán také plug-in do vývojových prostředí Eclipse a NetBeans. [36]

JavaFX Production Suite je sada nástrojů pro snadnější vytváření vzhledu aplikací. Obsahuje zásuvné moduly pro programy Adobe Photoshop a Adobe Ilustrátor, pomocí kterých lze snadno přenášet vytvořený návrh aplikace. [36]

JavaFX Desktop používá vlastností Java Platform Standart Edition (Java SE) 6 Update 10. Nabízí uživatelům možnost přesunutí spuštěné webové aplikace přímo na pracovní plochu. [36]

#### 6.1.2 Sun JavaFX 1.1

JavaFX 1.1 byla oficiálně vydána v únoru 2009. Jednalo se spíše o update, který odstraňoval řadu chyb z předcházející verze. Přinesl také vylepšení jazyka JavaFX Script - jedním z nich bylo například plná podpora primitivních typu z Javy (Long atd.). Největší novinkou této verze byla oficiální podpora JavaFX Mobile, což umožňovalo vytváření mobilních aplikací pomocí jazyka JavaFX Script. [37]

### 6.2 Oracle JavaFX 1.2

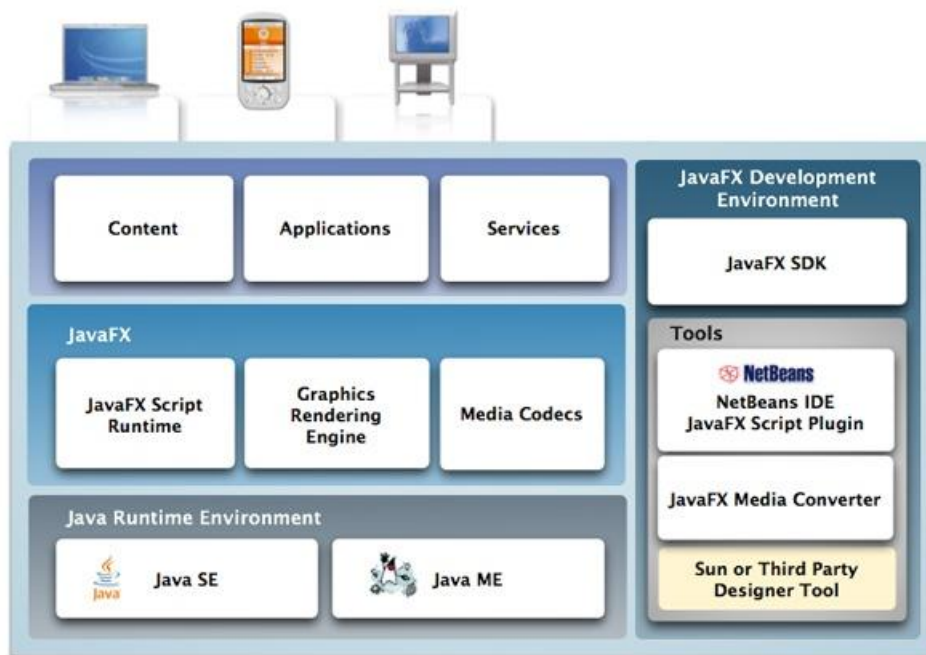
Nová verze JavaFX 1.2 byla představena již čtyři měsíce po předchozí verzi a to firmou Oracle, která koupila společnost Sun Microsystems. Odhalena byla na konferenci JavaOne, konaná v červnu 2009. Tato verze potěšila zejména příznivce systému Unix a po dlouhém čekání se uživatelé dočkali podpory pro operační systém Linux a OpenSolaris.

Pro vytváření JavaFX aplikací se používá skriptovací jazyk JavaFX Script, případně samotný jazyk Java. V obou jazycích lze vytvářet grafický návrh i funkcionalitu aplikace. Při psaní aplikací lze používat rozsáhlé knihovny například „java“, „javafx“.

Z pohledu tvorby aplikací byly přidány nové možnosti layoutu. V předcházejících verzích bylo možné vytvářet návrh aplikací jen pomocí obalových tříd „hbox“ (horizontální uspořádání) a „vbox“ (vertikální uspořádání). Do verze 1.2 byly pro snadnější návrh přidány čtyři nové obalové

třídy – „Flow“ (horizontální nebo vertikální zarovnání s ohledem na tvar objektů), „Stack“ (uspořádání odzadu dopředu), „Tile“ (dlaždicové uspořádání) a „ClipView“ (uspořádání s ohledem na obsah). [39]

Mezi další novinky, které přinesla tato verze, můžeme zařadit možnost skinování prvků pomocí kaskádových stylů a podporu RSS/Atom. Také byl přidán nový balíček Chart (pro vytváření a práci s grafy), třída pro výpočty (obdoba `java.lang.Math`) a třída pro definování vlastností (obdoba `java.util.Properties`). [40]



Obr. 6.1 Platforma JavaFX [41]

### 6.2.1 JavaFX Script

JavaFX Script vyvinul Chris Oliver a původně ho pojmenoval názvem F3 (Form Follows Function), na konferenci JavaOne 2007 byl název změněn. JavaFX Script je programovací jazyk na platformě Java, syntaxe je podobná JavaScriptu. Jazyk je jednoduchý, deklarativní a objektově orientovaný. [42]

Tento jazyk vznikl z důvodu omezení samotného jazyka Java. Při vytváření vzhledu aplikací pomocí Javy mohou programátoři využívat pouze Swing komponenty, případně Java 2D funkce. Jazyk JavaFX Script nabízí jednodušší a rychlejší způsob vytváření návrhu aplikace a není omezen pouze na Swing komponenty a 2D funkce. Také umožňuje navíc snadnější rozvržení prvků pomocí layoutů, vytváření animací, různé efekty s textem i grafikou, transformace s objekty apod. [43]

```

package javafxapplication;

import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.text.Text;
import javafx.scene.text.Font;

Stage {
    title: "Application title"
    scene: Scene {
        width: 250
        height: 80
        content: [
            Text {
                font : Font {
                    size : 16
                }
                x: 10
                y: 30
                content: "Application content"
            }
        ]
    }
}

```

**Výpis 6.1 Základní deklarace JavaFX**

Vytváření vzhledu aplikace v technologii JavaFX je značně odlišný od ostatních vybraných technologií. Struktura kódu není založena na XML formátu, ale je zde využíváno jiného formátu. Ve výpisu 6.1 je zobrazena základní struktura kódu. Na začátku dokumentu je vždy uvedeno, do kterého balíčku aplikace patří a jaké balíčky jsou importované k používání. Hlavním prvkem je „Stage“, v jeho atributech lze nastavit základní vlastnosti aplikace, jaké jsou popisek, výška, šířka aplikace atd. Klíčovým atributem tohoto prvku je „scene“, ten upřesňuje, jaká scéna bude zobrazena. Scéna je prvek, do kterého se vkládají grafické prvky.

Vzhled aplikace v jazyce JavaFX Script lze vytvářet dvěma způsoby. Prvním je jednoduché přetažení prvku z nabízené palety a následná úprava jeho vlastností. Druhou možností je prvek vytvořit a nadefinovat. Prvky lze vytvářet také přímo v Java kódu, jak je zobrazeno ve výpisu 6.4.

```

import javafx.ext.swing.SwingButton;
SwingButton {
    name: "Button1"
    text: "Tlačítko"
    action: Akce
}

```

**Výpis 6.2 Vytvoření tlačítka v jazyce JavaFX (přetažení z palety)**

```
import javafx.ext.swing.SwingButton;
var Button1:SwingButton = new SwingButton();
Button1.text = "Tlačítko";
Button1.action = Akce;
```

Výpis 6.3 Nadefinování tlačítka v jazyce JavaFX

```
import javax.swing.JButton;
import java.awt.event.ActionListener;
JButton Button1 = new JButton("Tlačítko");
Button1.addActionListener(actionListener);
```

Výpis 6.4 Vytvoření tlačítka v jazyce Java

### 6.2.2 Java

Java je objektově-orientovaný programovací jazyk vytvořený v roce 1995 společností Sun Microsystems. Považuje se za jeden z nejpoužívanějších programovacích jazyků na světě.

Java je interpretovaný jazyk - nevytváří se skutečný strojový kód, ale pouze mezikód, který je nezávislý na zařízení. Pro běh aplikace napsané v Javě je zapotřebí mít na daném zařízení virtuální stroj Javy. [44]

Syntaxe jazyka Java je zjednodušenou verzí syntaxe jazyka C a C++. Z jazyka odpadla většina konstrukcí, naopak byla přidána řada nových rozšíření. Mezi jeho základní vlastnosti patří bezpečnost, přenositelnost, výkonnost, podpora zpracování vícevláknových aplikací a nezávislost na zařízení. Právě kvůli těmto vlastnostem se Java používá pro vytváření robustních a vysoce spolehlivých aplikací. [44]

```
package helloworld;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.VBox;
import javafx.ext.swing.SwingLabel;
import javafx.ext.swing.SwingButton;
import javafx.geometry.HPos;

var helloWorld:String;

Stage {
  scene: Scene {
    width: 150
    height: 100
    content: [
      VBox {
        width: 150
        height: 100
        layoutY: 50
        nodeHPos: HPos.CENTER
```

```

hpos: HPos.CENTER
content: [
    SwingLabel {
        text: bind helloWorld
    }
    SwingButton {
        text: "Click Me"
        action: function() { helloWorld = "Hello World" }
    }
]
}
}
}
}
}

```

Výpis 6.5 Ukázková JavaFX aplikace Hello World

### 6.2.3 Vývojové prostředí

V současné době lze JavaFX aplikace vyvíjet pomocí JavaFX SDK nebo prostřednictvím instalace přídatného modulu do vývojového prostředí.

JavaFX SDK obsahuje Desktop Runtime, Mobile Emulator (pouze pro Windows), JavaFX API, kompilátor a dokumentaci. Jedná se o command-line nástroj, tedy vše se ovládá pomocí příkazového řádku. Je dostupný pro operační systémy Windows, Mac OS, Linux a Solaris (x86). [45]

Přídavný modul pro současnou verzi JavaFX 1.2 existuje pro vývojové nástroje NetBeans a Eclipse. Oba nástroje jsou pod open-source hlavičkou a jsou k dispozici zdarma. Po nainstalování pluginu je k dispozici editor, paleta nástrojů, navigátor a náhled aplikace. Editor podporuje zvýraznění klíčových slov, obsahuje funkci doplňování kódu a nápovědu.

Grafický návrh lze vytvořit za použití JavaFX Production Suite (plugin pro grafické programy Adobe Photoshop a Adobe Illustrator). Vytvořený návrh aplikace ve zvoleném grafickém programu lze pomocí přidaného modulu uložit do JavaFX souboru FXZ nebo FXD. [38]

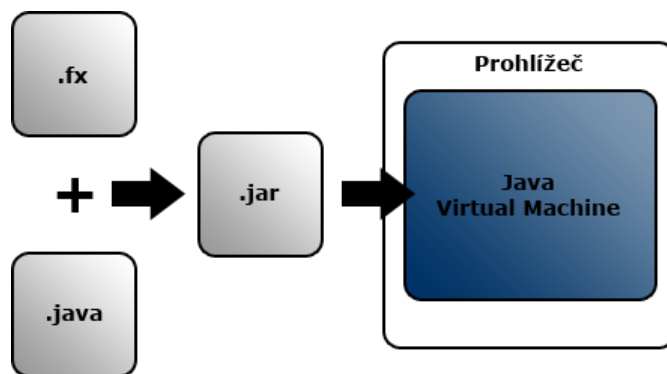
Pro snadnější návrh vzhledu aplikace existují ještě dva nástroje – JavaFX Pad a JavaFX Builder. JavaFX Pad je velmi jednoduchý editor, který nabízí pouze vizuální náhled napsaného kódu. Druhý jmenovaný nástroj je JavaFX Builder, vytvořený firmou ReportMill Software s podporou společnosti Sun Microsystems. Jedná se o plnohodnotný grafický program pro JavaFX. Nabízí snadné vytváření scény pomocí vkládání jednotlivých prvků z palety a následnou možností je graficky upravit. Umožňuje také vytváření různých tvarů, k dispozici je obdélník, elipsa, polygon, hvězda, případně lze použít i nástroj tužka. Mimo základních komponent nabízí také možnost přidání tabulek, grafů s možností připojení k databázi. JavaFX Builder je také zatím jediný nástroj, který umožňuje snadnější vytváření animací.

### 6.2.4 Běhové prostředí

JavaFX aplikace lze spouštět přes webový prohlížeč, přímo z plochy počítače nebo na mobilním zařízení. Pro běh aplikací na daném zařízení je vyžadováno nainstalovat

virtuální stroj Javy - Java Virtual Machine (JVM). Ten je dostupný v instalačním balíčku Java Development Kit (JDK) nebo Java Runtime Environment (JRE), případně Java Standard Edition (Java SE). Spuštění aplikace ve webovém prohlížeči vyžaduje také instalaci Java Plug-in přímo do prohlížeče. Technologie JavaFX je podporována od verze 1.5.

Pro nasazení samotné desktop aplikace slouží technologie Java Web Start (JWS). Ta umožňuje spuštění aplikací stažených z internetu nebo uložených na lokálním disku, také jejich správu a případné vytvoření zástupce na plochu. Spouštěcí soubor má koncovku JNLP, jedná se o klasický XML formát. Při spuštění souboru se přes webový prohlížeč předají potřebné informace pro JWS, ten se postará o stažení, instalaci a spuštění aplikace. [46]



Obr. 6.2 Struktura JavaFX aplikace

### 6.2.5 Nasazení aplikace do webové stránky

Vložení JavaFX aplikace do stránky je velice jednoduché. Do HTML značky „script“ je vložen prvek „javafx“, který obsahuje základní údaje pro spuštění dané aplikace. Pokud se jedná o webovou aplikaci, data se načítají z XML souboru „nazevAplikace\_browser.jnlp“. Ten obsahuje údaje o aplikaci (název, autor, popis a místo uložení aplikace) a informaci o potřebné nejvyšší verzi JVM, ve kterém lze danou aplikaci spustit. Pro správné fungování aplikace je potřeba nastavit v souboru JNLP webovou adresu umístění aplikace, tento údaj je nutné zadat do atributu „codebase“ elementu „jnlp“ a také do „href“ elementu „homepage“.

```

<script>
  javafx(
    {
      archive: "HelloWorld.jar",
      draggable: true,
      width: 200,
      height: 200,
      code: "helloworld.Main",
      name: "HelloWorld"
    }
  );
</script>

```

Výpis 6.6 HTML kód JavaFX aplikace vkládaný do webové stránky (ukázka Hello World)



```

<?xml version="1.0" encoding="UTF-8"?>
<jnlp spec="1.0+" codebase="" ">
  <information>
    <title>HelloWorld</title>
    <vendor>Baran</vendor>
    <homepage href="" "/>
    <description>HelloWorld</description>
    <offline-allowed/>
    <shortcut>
      <desktop/>
    </shortcut>
  </information>
  <resources>
    <j2se version="1.5+"/>
    <extension name="JavaFX Runtime" href="http://dl.javaafx.com/1.2/javaafx-rt.jnlp"/>
    <jar href="HelloWorld.jar" main="true"/>
  </resources>
  <applet-desc name="HelloWorld" main-class="com.sun.javaafx.runtime.adapter.Applet" width="200"
height="200">
    <param name="MainJavaFXScript" value="helloworld.Main">
  </applet-desc>
  <update check="background">
</jnlp>

```

**Výpis 6.7 Kód souboru HelloWorld\_browser.jnlp**

### 6.3 Budoucnost

Na konferenci Devovx 2009, která byla uskutečněna v prosinci 2009 v Belgii, byly zveřejněny novinky pro nadcházející verzi JavaFX 1.3. V prezentaci věnované JavaFX byly předvedeny plánované grafické prvky pro nadcházející verze. Je tedy více než pravděpodobné, že ve verzi 1.3 přibudou nové grafické prvky, mezi které mohou patřit například TreeView, PasswordBox, Menus, ToolBar, Scroll Bar, Tooltip. [47]

Sun Microsystem představil na konferenci JavaOne 2009 nový nástroj JavaFX Authoring Tool. Tento nástroj by měl přinést řadu usnadnění pro tvorbu JavaFX aplikací. Jednou ze schopností, kterou by měl umožňovat, je sjednocení vzhledu aplikace vytvořeného v grafickém programu a kódu vytvořeného ve vývojovém prostředí. Nový nástroj by měl také zvládat vytváření animací pomocí klíčových snímků, obdobně jako u nástroje Adobe Flash. [48]

## 7 Porovnání vlastností vybraných technologií

V této části své práce bych rád shrnul základní vlastnosti vybraných technologií. Následně porovnám dané konkurenční technologie z pohledu vývojových nástrojů a rozšířenosti běhového prostředí.

### 7.1 Porovnání základních vlastností

V tabulce 7.1 lze vidět souhrn základních informací o jednotlivých technologiích. Každá z těchto technologií umožňuje vytvářet grafické rozhraní. Sada grafických prvků se u vybraných technologií výrazně neliší, všechny obsahují základní škálu ovládacích prvků (TextBox, Button, ProgressBar atd.) a prvků pro návrh a rozvržení aplikace (Panel, Tile atd.). Flex a JavaFX navíc nabízejí ve své paletě komponenty pro vytváření grafů.

	Adobe Flex	Microsoft Silverlight	JavaFX
<b>Verze</b>	3.0	3.0	1.2
<b>Jazyky</b>	MXML ActionScript 3	XAML C# / VB.NET	JavaFX Script Java
<b>Výsledný soubor</b>	.swf	.xap (ZIP formát)	.jar (ZIP formát)
<b>Běhové prostředí pro webový prohlížeč</b>	Adobe Flash Player 9 a vyšší	Silverlight 3 Plug-in	Java Plug-in 1.5 a vyšší
<b>Vytváření grafického uživatelského rozhraní</b>	Ano	Ano	Ano
<b>Animace</b>	Ano (složitější animace pomocí Flash - frame-based model)	Ano (WPF animační model)	Ano (frame-based model)

Tab. 7.1 Souhrn základních vlastností vybraných technologií

### 7.2 Animace

Porovnání z pohledu vytváření animací Adobe Flex pokulhává vůči ostatním technologiím. Lze v něm vytvářet pouze jednoduché přechodové animace, pro složitější tvorbu animací je doporučen nástroj Adobe Flash. Ten je založen na frame-based modelu, princip tohoto modelu spočívá ve vytváření klíčových snímků a přechodů mezi nimi.

Animace vytvářené v Silverlightu jsou založené na WPF a využívají time-based modelu. Nevytváří se jednotlivé snímky animace, ale stačí definovat typ a délku operace. Silverlight si podle definovaných údajů dopočítá interpolaci mezilehlých hodnot a rychlost přehrávání snímků.

JavaFX umožňuje vytvářet jednoduché i složitější animace. Stejně jako Adobe Flash je založen na frame-based modelu. Značnou nevýhodou při tvorbě animací ve vývojových prostředích je nutnost vytvářet animace pomocí animačních prvků. Pro jednodušší tvorbu lze použít program JavaFX Builder, který umožňuje vytvářet animace a současně generovat JavaFX kód dané animace.

### 7.3 Velikost výsledného souboru

Za povšimnutí stojí formát výsledného souboru. U technologií Silverlight a JavaFX je výsledný soubor komprimován do ZIP formátu, tím je dosaženo lepších výsledků velikosti konečného souboru. Flex SWF formát není nikterak komprimován a velikost souboru v debug verzi je větší než u dvou výše zmiňovaných technologií. Release verze má o poznání menší velikost; důvodem je odstranění informací potřebných pro ladění.

Pro porovnání velikosti výsledného souboru jsem použil ukázkový příklad „HelloWorld“ (soubory jsou přiložené v příloze č. 1). Výsledné velikosti souborů v debug a release verzi lze vidět v tabulce 7.2.

Nejmenší velikost souboru má technologie JavaFX. Soubor JAR obsahuje pouze meta informace a soubory CLASS, proto konečná verze dosahuje až neuvěřitelně malé velikosti. U technologie Silverlight je již velikost souboru podstatně větší a je závislá na počtu a velikosti přiložených knihoven používané v dané aplikaci. U Silverlight lze také vidět rozdíl velikosti mezi použitým programovacím jazykem. Při použití jazyka C# byla vyžadována navíc knihovna Controls. Flex aplikace dosáhla ve vybraných technologiích největší velikosti výsledného souboru, důvody již byly vypsány výše.

	Flex (.swf)	Silverlight C# (.xap)	Silverlight VB.NET (.xap)	JavaFX (.jar)
<b>Debug verze</b>	248 kB	148 kB	53 kB	7 kB
<b>Release verze</b>	153 kB	148 kB	53 kB	4 kB

Tab. 7.2 Velikost výsledných souborů ukázkového příkladu "HelloWorld"

### 7.4 Porovnání možností vývojových prostředí

Pro vytváření Flex aplikací existuje, v předešlých částech již zmiňovaný, Adobe Flex Builder 3. Oproti ostatním technologiím existuje pro Flex pouze toto vývojové prostředí. Lze to považovat za výhodu, jelikož Adobe Flex Builder 3 obsahuje všechny důležité nástroje pro vývoj aplikací – vizuální návrh, nástroje pro analýzu a ladění apod. Je tedy možno pohodlně vytvořit celou aplikaci v tomto prostředí, ostatní vybrané technologie vyžadují mnohdy využití více vývojových nástrojů. Flex Builder je placený nástroj a cena se v dnešní době pohybuje v řádech tisíců korun. Další nevýhodou nástroje je již zmiňované omezení vytváření animací.

Pro vývoj Silverlight aplikací existují celkem tři vývojová prostředí, bohužel ani jeden z těchto programů nenabízí kompletní balík nástrojů. Microsoft Visual Web Developer 2008 Expression Edition SP1 obsahuje pouze základní možnosti - vzhled aplikace lze psát ručně nebo vkládat prostřednictvím panelu komponent. Vývojové prostředí Microsoft Visual Studio 2008 SP1 oproti předcházejícímu prostředí obsahuje navíc nástroje pro ladění. Toto prostředí existuje také ve verzi Express Edice, která je stejně jako Visual Web Developer 2008 Expression Edition zdarma. Poslední nástroj, který lze využít pro vývoj Silverlight aplikací je Microsoft Expression Blend. Tento nástroj se zejména používá pro návrh vzhledu aplikace a tvorbu animací. Obsahuje vizuální editor s řadou možností a efektů pro úpravu vzhledu komponent. Expression Blend je využíván výhradně pro návrh, jelikož postrádá jakékoliv nástroje pro ladění. Při vytváření

Silverlight aplikací lze snadno přepínat mezi vývojovými prostředími Express a Visual Studio 2008 (oba nástroje musí být nainstalovány na daném zařízení). Je možné tak dosáhnout větší efektivity práce. Vývojové prostředí Visual Studio 2008 a Expression Blend jsou placené.

JavaFX aplikace se nejčastěji vytvářejí ve vývojových prostředích NetBeans a Eclipse. Obě prostředí obsahují stejnou škálu komponent a nabízejí obdobné nástroje. Ovšem ani jeden ze základních nástrojů neumožňuje vizuální návrh aplikací. Pro prostředí NetBeans je v současné době dostupný plugin s názvem JavaFXComposer. Ten umožňuje mimo vizuálního návrhu také vytváření jednoduchých animací, případně jednoduchý přístup k datům. Pro prostředí Eclipse je firmou Exadel vyvíjen plugin JavaFX Studio, který by měl v budoucnu rovněž umožňovat vizuální návrh. Pro návrh aplikace lze také použít grafický program JavaFX Builder, který vytvářený vzhled generuje přímo do jazyka JavaFX Script. Další možnost je využít JavaFX Production Suite obsahující přídavné moduly do grafických programů Adobe Photoshop a Adobe Illustrator (oba programy jsou placené). Výhodou, která hraje pro JavaFX, je cena vývojových prostředí NetBeans, Eclipse i JavaFX Builder. Všechny zmíněné programy jsou zdarma a tak lze považovat vývoj JavaFX aplikací za nejlevnější.

	Adobe Flex	Microsoft Silverlight			JavaFX	
<b>Vývojový nástroj</b>	Adobe Flex Builder 3 (Eclipse)	Microsoft Expression Blend	Microsoft Visual Studio 2008 SP1	Microsoft Visual Web Developer 2008 Expression Edition SP1	NetBeans 6.8 s JavaFX	Eclipse IDE for Java EE Developers (Eclipse Plugin)
<b>Vizuální editor</b>	Ano	Ano	Ne	Ne	Ano (JavaFXComposer)	Ne
					(JavaFX Builder, JavaFX Production Suite)	
<b>Panel s nabídkou komponent</b>	Ano	Ano	Ano	Ano	Ano	Ano
<b>Nástroje pro ladění</b>	Ano	Ne	Ano	Ano (omezené možnosti)	Ano	Ano
<b>Zvýrazňování klíčových slov</b>	Ano	Ano	Ano	Ano	Ano	Ano
<b>Placený nástroj</b>	Ano (Studentská verze zdarma - ne pro komerční účely)	Ano	Ano (Expression verze zdarma)	Ne	Ne	Ne

Tab. 7.3 Souhrn vlastností vývojových prostředí pro dané technologie

## 7.5 Rozšířenost běhového prostředí

Z pohledu rozšířenosti běhového prostředí je na tom nejlépe technologie Adobe Flex. V tabulkách 7.4 a 7.5 lze vidět, že běhové prostředí Adobe Flash Player je nainstalováno na většině počítačů. A to bez rozdílu na prohlížeč nebo operační systém. Z tabulky lze také vyčíst, že běhové prostředí pro Silverlight technologii se na operačním systému Linux vyskytuje pouze ve 4,6 %, což je oproti ostatním technologiím velmi málo.

Zjištěná hodnota běhové prostředí Java Plug-in pro webový prohlížeč Opera je skoro nulová. Tato hodnota je ovlivněna zejména skutečností, že Opera od verze 4 má vestavěnou Java podporu využívající Java Runtime Environment nainstalovaný na daném počítači, nevyužívá tedy žádný plugin.

V tabulce zaměřené na porovnání výskytu běhových prostředí podle operačního systému si lze všimnout, že Java Plug-in verze 1.6 se na operačním systému Windows 7 vyskytuje pouze v 0,6 %. Důvodem tak nízké hodnoty je pozdější podpora tohoto pluginu pro Windows 7. Přidání podpory pro tento operační systém bylo až ve verzi Java SE 6 Update 18, která byla vydána 14. ledna 2010. Pro přechod na novou verzi je vyžadováno odinstalování předcházející verze.

	Mozilla Firefox		Internet Explorer		Opera		Google Chrome		Safari	
Adobe Flash Player 9	6,4	98,6	7,0	97,1	12,6	76,5	4,7	98,8	6,9	82,8
Adobe Flash Player 10	92,2		90,1		63,9		94,1		75,9	
Silverlight 3 Plug-in	55,1		50,6		-		55,9		30,8	
Java Plug-in 1.5	8,5	78,9	11,5	64,9	0,0	0,4	-	69,1	57,4	82,9
Java Plug-in 1.6	70,3		53,4		0,4		69,1		25,5	
Java Plug-in 1.7	0,1		-		-		-		-	

Tab. 7.4 Rozšířenost běhového prostředí v závislosti na webovém prohlížeči (v procentuálním vyjádření) [49]

	WIN XP		WIN Vista		WIN 7		Linux		MAC OSX	
Adobe Flash Player 9	7,2	97,2	4,9	98,6	0,2	78,5	21,9	96,1	8,0	86,9
Adobe Flash Player 10	90,0		93,7		78,3		74,2		78,9	
Silverlight 3 Plug-in	43,0		63,2		58,4		4,6		31,2	
Java Plug-in 1.5	17,4	62,9	1,9	78,6	79,9	80,5	10,3	56,9	44,5	88,3
Java Plug-in 1.6	45,5		76,7		0,6		44,1		43,8	
Java Plug-in 1.7	-		-		-		2,5		-	

Tab. 7.5 Rozšířenost běhového prostředí v závislosti na operačním systému (v procentuálním vyjádření) [49]

## 8 Porovnání z programátorského hlediska

Tato kapitola diplomové práce se zabývá porovnáním vybraných technologií z pohledu jednoduchosti vývoje aplikací. Abych mohl provést porovnání, rozhodl jsem se vytvořit aplikaci, která bude stejná pro všechny technologie. V dnešní době se RIA aplikace nejčastěji využívají pro práci s médii (video, audio), z toho důvodu jsem se rozhodl vytvořit webový videopřehrávač s možností spouštět videa umístěné na internetu.

### 8.1 Tvorba videopřehrávače

Dnes se webové videopřehrávače používají na přehrávání videí s různým zaměřením, nejvíce se vyskytují na publicistických nebo zábavných serverech. Přehrávače umožňují základní funkce jako spuštění, pozastavení a zastavení videa. Také nabízejí možnost nastavení hlasitosti a posouvání nahrávky videa. Některé videopřehrávače navíc mají schopnost zobrazit videa na celou obrazovku, přepnutí videa do lepšího formátu apod.

Mým cílem bude ve vybraných technologiích vytvořit obdobný videopřehrávač, a to jak funkcionálně tak vzhledově. Videopřehrávač bude načítat médium z internetu pomocí zadané webové adresy. Funkce, které bude přehrávač nabízet, bude přehrávání, pozastavení a zastavení videa. Dále možnost nastavení hlasitosti zvuku, případně zvuk zcela vypnout. Přehrávač bude také zobrazovat průběh videa s možností skoku na jeho libovolnou část.



Obr. 8.1 Grafický návrh přehrávače

Pro implementaci videopřehrávače jsem zvolil pro mě dostupné vývojové prostředí. Volba nástroje pro Flex byla jednoznačná, jelikož existuje pouze jedno vývojové prostředí a to právě Adobe Flex Builder 3. Výběr prostředí pro Silverlight byl mírně komplikovanější, a proto jsem vybral ideální kombinaci dvou nástrojů - MS Visual Studio 2008 a MS Expression Blend.

Videopřehrávač v technologii JavaFX jsem vytvářel v NetBeans ve verzi 6.8, toto vývojové prostředí mi osobně vyhovuje více než Eclipse.

### 8.1.1 Podpora video a audio médií

Jednou z nejdůležitějších věcí u přehrávačů je škála podporovaných formátů. Nejrozmanitější škálu má technologie Silverlight. V tabulce níže lze vidět všechny formáty, které jsou ve vybraných technologiích podporovány. Technologie JavaFX používá pro přehrávání audio i video formátů kodeky dostupné na daném operačním systému.

	Video formáty	Audio formáty
<b>Adobe Flex 3</b>	MP4 Soubor FLV (Sorenson Spark) Soubor FLV (ON2 VP6)	MP3 AAC+ / HE-AAC / AAC v1 / AAC v2 Nellymoser Speex
<b>Microsoft Silverlight 3</b>	Raw Video MP4 RGBA WMV1, WMV2, WMV3, WMVA, WVC1 YV12 QuickTime	MP3 AAC / AAC-LC / HE-AAC WAV WMA Standart/Professional
<b>JavaFX 1.2</b>	<b>Windows</b> : Video formáty spustitelné ve Windows Media Playeru (ASF/ WMV / AVI / MPEG / MPEG-1)  <b>MAC OS X</b> : Video formáty podporované Core Video (MOV / MP4 / MPEG)  <b>Ubuntu, Solaris</b> : Video formáty podporované Gstreamer (ASF/ WMV / WM/ AVI / MPEG / MPEG-1)	MP3 <b>Windows</b> : Audio formáty spustitelné ve Windows Media Playeru (VMA / WAV / MIDI) <b>MAC OS X</b> : Audio formáty spustitelné v QuickTime (MP2 / AIFF / WAV / M4A/ M4B / M4P / MIDI)  <b>Ubuntu, Solaris</b> : Audio formáty podporované Gstreamer (VMA / WAV / MIDI)

Tab. 8.1 Podpora formátů ve vybraných technologiích

### 8.1.2 Komponenty pro přehrávání videí

Ve zvolených technologiích existuje již vytvořena komponenta pro práci s médii. V jednotlivých částech budou popsány podrobněji.

#### 8.1.2.1 Adobe Flex

Technologie Flex nabízí pro práci s videem komponentu „videoDisplay“. Ta podporuje postupné stahování souboru přes HTTP protokol nebo streamování z Flash Media Serveru. Mezi

hlavní vlastnosti, které tato komponenta nabízí, patří připojení zdroje souboru, zjištění celkového času média, nastavení hlasitosti zvuku apod. Samozřejmostí jsou metody pro načtení, přehrání, pozastavení a zastavení videa. Tato komponenta nabízí řadu nastavení a využití.

```
<mx:VideoDisplay id="Video" x="0" y="0" width="400" height="300"
complete="Video_MediaEnded(event)" playheadUpdate="Video_playheadUpdate(event)"
mouseOver="Video_MouseOver(event)" stateChange="checkForErrors(event)" autoPlay="false"/>
```

Výpis 8.1 Komponenta „videoDisplay“ ve vytvořené aplikaci VideoPlayer (Adobe Flash)

### 8.1.2.2 Microsoft Silverlight 3

Pro přehrávání videa nabízí Silverlight celkem dvě komponenty. První z nich je „MediaPlayer“ a jedná se o již vytvořený přehrávač. Uživatel jednoduše nastaví základní vlastnosti komponenty a může videopřehrávač používat. Zvolil jsem pro tvorbu přehrávače druhou komponentu označovanou názvem „MediaElement“. Jedná se o obdélníkovou oblast, která umožňuje přehrávání audia a videa. Jedná se o UIElement, proto podporuje také vstup z myši a klávesnice. Pro připojení média se používá vlastnost „source“. Stejně jako „videoDisplay“ u technologie Flex podporuje „MediaElement“ základní metody pro práci s videem.

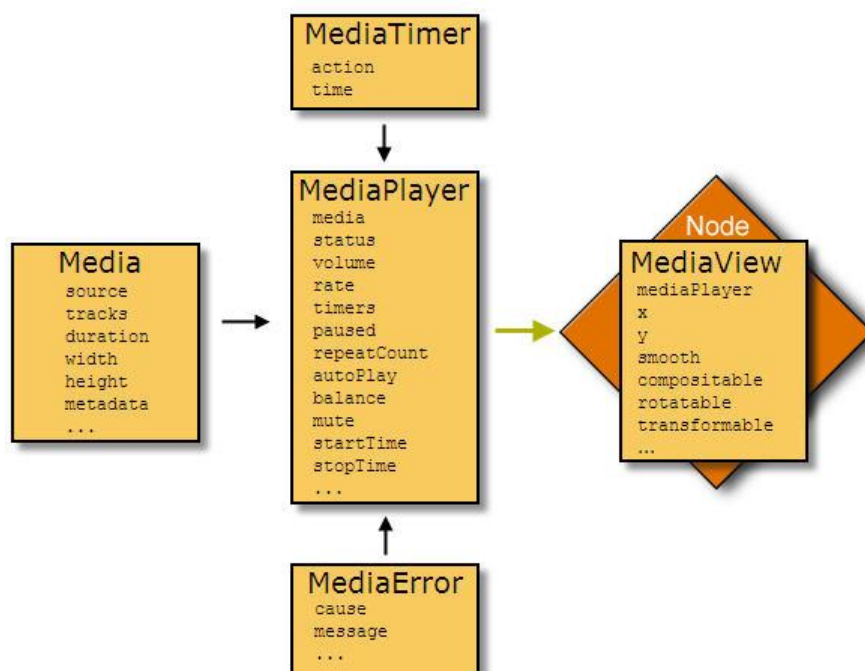
```
<MediaElement x:Name="Video" DownloadProgressChanged="Video_DownloadProgressChanged"
MediaEnded="Video_MediaEnded" MouseEnter="Video_MouseEnter"/>
```

Výpis 8.2 Komponenta „MediaPlayer“ ve vytvořené aplikaci VideoPlayer (Microsoft Silverlight)

### 8.1.2.3 JavaFX

Komponentu pro přehrávání videa lze v Silverlightu a Flexu jednoduše vložit do aplikace a následně nastavit podle atributů; v porovnání s nimi je v technologii JavaFX přidání media přehrávače složitější. JavaFX Media API obsahuje celkem pět tříd (viz obr. 8.2). Pro vytvoření funkčního videopřehrávače je zapotřebí využít nejméně tří tříd, kterými jsou „MediaView“, „MediaPlayer“ a „Media“. „MediaView“ je podtřídou kořenové třídy Node a podporuje animaci, průsvitnost a různé efekty. Další třídou je samotný „MediaPlayer“, který slouží pro přehrávání video nebo audio souborů. „MediaPlayer“ potřebuje pro přehrání daného videa třídu „Media“. Tato třída reprezentuje zdroj souboru a obsahuje informace o daném mediálním souboru např. metadata, rozlišení atd. Pro odchyčení chyb, které nastanou při načítání media, lze použít třídu „MediaError“. Poslední z použitelných tříd je „MediaTimer“, ta umožňuje specifikovat čas a akci, která má být provedena po dosažení zadaného času při přehrávání.





Obr. 8.2 JavaFX Media API [50]

```

var video: MediaPlayer = MediaPlayer {
    volume: bind (1 - sVolume.value)
    onEndOfMedia: video_mediaEnded
}
var mediaView: MediaView = MediaView {
    preserveRatio: true
    mediaPlayer: bind video
    fitHeight: bind height
    fitWidth: bind width
    onMouseEntered: application_mouseEntered
    onMouseExited: application_mouseExited
}

```

Výpis 8.3 „MediaPlayer“ a „MediaView“ ve vytvořené aplikaci VideoPlayer (Oracle JavaFX)

## 8.2 Porovnání technologií z pohledu programování

Po velkém úsilí se mi nakonec podařilo vytvořit skoro autentické videopřehrávače. Vybrané technologie nabízejí v některých případech odlišné možnosti programování, a tudíž se v některých detailech přehrávače liší.

Pokud mám porovnat technologie z pohledu programování, nejlépe se mi programovaly Silverlight aplikace. Důvodem není pouze předešlá znalost programovacího jazyka C#, ale zejména vlastnosti a principy, které tento jazyk nabízí. Vývojové nástroje umožňují řadu usnadnění pro programátory; například automatické generování metod pro dané akce. Nástroje také nabízejí obsáhlý IntelliSense s nápovědou; není tedy potřeba hledat tyto informace v nápovědě. Programování Silverlight aplikací bylo ze zvolených technologií snadné a jednoduché. Nevýhodou

bylo přepínání mezi vývojovými prostředími, jelikož jsem pro vytvoření videopřehrávače používal Expression Blend a Visual Studio 2008.

Technologie Flex, přesněji jazyk ActionScript, je podle mého názoru z hlediska efektivity programování na druhém místě. Důvodem tohoto umístění jsou nedostatečné možnosti pro usnadnění programování ve vývojovém prostředí Adobe Flex Builder 3, přesto syntaxe ActionScriptu mi připadá jednodušší a snadněji naučitelná než jazyk C# nebo JavaFX. Flex Builder 3 obsahuje pouze jednoduchý IntelliSense, proto je v některých případech nutné nahlédnout do nápovědy.

Technologie JavaFX byla pro mě úplně nová. Při prvním kontaktu jazyka JavaFX Script, mi připadala jeho struktura značně komplikovaná. Zároveň jsem neobjevil žádnou podobnost se známějšími programovacími jazyky. Vytváření aplikace v této technologii připomíná „skládačku“, kde se všechny předem vytvořené a nadefinované prvky skládají dohromady. Pro tvorbu přehrávače jsem použil vývojové prostředí NetBeans. Ačkoliv obsahuje funkce pro snadnější vytváření aplikací je vytváření velice zdlouhavé.

## 9 Výkonnostní testy

Jedním z cílů této práce je porovnat vybrané technologie z pohledu výkonu. K tomuto účelu jsem vytvořil dva výkonnostní testy. První test se zabývá rychlostí vykreslování grafických prvků. Druhý test je zaměřen na rychlost zpracování složitějšího algoritmu.

Aplikace vytvořené pro výkonnostní testy jsem testoval ve třech nejpoužívanějších prohlížečích – Internet Explorer, Mozilla Firefox a Google Chrome. Použil jsem nově nainstalované prohlížeče, aby nedošlo k ovlivnění výsledků. Ve všech zvolených prohlížečích jsou nainstalované následující zásuvné moduly Adobe Flash Player ve verzi 10.0.45.2 a Silverlight Plug-in 3.0.50106.0. Pro technologii JavaFX obsahuje prohlížeč Internet Explorer plugin Java Plug-In 2 SSV Helper verze 6.0.180.7, Firefox a Chrome používají Java Plug-in 1.6.0\_18 for Mozilla browsers.

Aplikace byly umístěny na server aspone.cz, přesněji na freehosting. Na serveru běží operační systém Windows Server 2008 s podporou všech vybraných technologií.

Oba testy probíhaly na notebooku Hewlett-Packard 550 s následujícími specifikacemi: dvoujádrový procesor C2D T5470 (1.6 GHz), paměť 2 GB RAM, integrovaná grafická karta Intel 965 Express. Na zařízení byl nainstalován operačním systémem Windows 7 Professional (32 bitová verze).

### 9.1 Test rychlosti načítání a vykreslování grafických prvků

Test je zaměřen na rychlost načítání aplikace a následné vykreslení grafických prvků. Pro testování jsem vytvořil jednoduché aplikace a zvolil jsem prvky (button, label, checkbox a textbox), které jsou dostupné ve všech třech technologiích. Ve zdrojovém kódu aplikace je zadán požadovaný počet prvků, které jsou při spuštění aplikace pomocí algoritmus náhodně vygenerovány. Pro své testovací účely jsem zvolil počet 100, 500, 1000, 1500, 2000, 2500 a 5000. Tento výběr považuji za vhodný a myslím, že tento počet prvků umožní dostačující porovnání vybraných technologií.

#### 9.1.1 Způsob měření

Původní myšlenkou bylo pro účel testování použít testovací software. Bohužel se mi nepodařilo najít žádný vhodný, který by podporoval všechny vybrané technologie. Na doporučení jsem vyzkoušel doplněk do prohlížeče Mozilla Firefox jménem YSlow, který informuje o rychlosti načtení stránky. Nevýhodou tohoto doplňku je omezení pouze na prohlížeč Firefox. Také jsem se setkal s problémem pro Flex aplikaci, kdy načtení stránky proběhlo dříve, než byly grafické prvky zobrazeny.

Uvažoval jsem nad způsobem měření rychlosti načítání a vykreslování grafických prvků testovaných aplikací ve více webových prohlížečích. Nakonec jsem se rozhodl pro využití JavaScriptu ve webové stránce. Měření probíhá způsobem porovnávání času před a po načtení a vykreslení grafických prvků. Čas před načtením aplikace je pořízen pomocí JavaScriptu následně při načítání stránky. Zjištění času pro načtení a zobrazení prvků je v technologii Flex zajištěno pomocí metody „applicationComplete“. Silverlight nabízí událost „CompositionTarget.Rendering“, která zajišťuje opakované vykreslování obsahu aplikace. V JavaFX jsem použil funkci „run“, kde

se nejprve zobrazí scéna a následně je volána metoda pro výpočet celkového času. Výsledky ve všech aplikacích jsou zobrazovány pomocí informačního okna.

### 9.1.2 Výsledky testování

Naměřené hodnoty rychlosti načítání a vykreslení aplikací jsou zobrazeny v tabulce 9.1. Nejlepších výsledků v tomto testu dosáhla technologie Adobe Flex, která ve většině případů zobrazila scénu nejrychleji. Druhá skončila technologie Silverlight a větších časových rozdílů oproti Flexu si lze všimnout zejména při zpracování většího počtu grafických prvků.

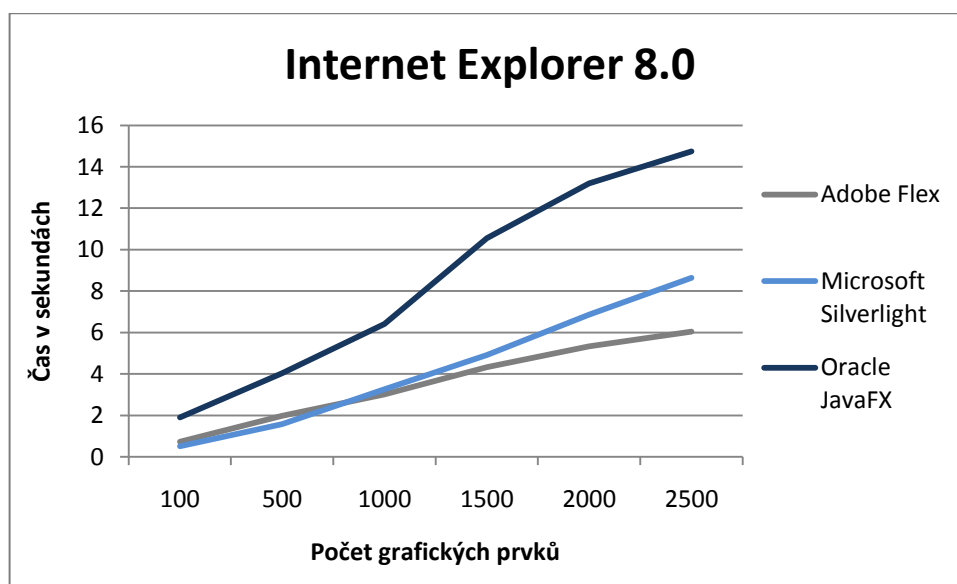
Nejhůře dopadla technologie JavaFX, která v naměřených hodnotách dosahuje až extrémního času. Nutno dodat, že naměřené hodnoty jsou již se spuštěným běhovým prostředím, samotné spouštění trvá dalších 3 - 5 sekund. Při počtu nad 1000 grafických prvků bylo nutné pro pokračování testování restartovat běhové prostředí (tedy internetový prohlížeč), jelikož se následující aplikace s větším počtem prvků již nedokázala zobrazit. Se zobrazením 5 000 grafických prvků si technologie JavaFX nedokázala poradit vůbec.

Z naměřených hodnot je možné porovnat také výkon jednotlivých pluginů v závislosti na webovém prohlížeči. Nejlépe si při testování vedl prohlížeč Google Chrome, který dosáhl nejlepších časů při načítání a zobrazování Flex a Silverlight aplikací. Zpracování JavaFX aplikací ve většině případů nejlépe zvládl prohlížeč Mozilla Firefox. Za zmínku stojí, že znatelně horšího výkonu oproti ostatním prohlížečům dosáhl Firefox při načítání a zobrazování Silverlight aplikací.

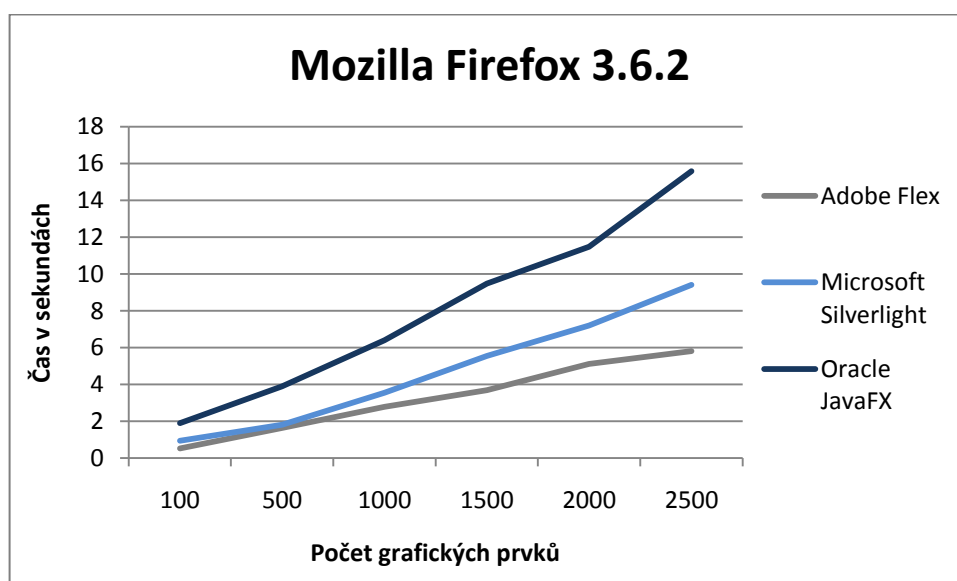
		Počet prvků						
		100	500	1000	1500	2000	2500	5000
Internet Explorer 8.0	Adobe Flex	724	1973	3007	4334	5328	6045	13765
	MS Silverlight	505	1569	3258	4910	6859	8639	21427
	Oracle JavaFX	1893	4040	6408	10561	13204	14743	-
Mozilla Firefox 3.6.2	Adobe Flex	516	1638	2785	3692	5109	5801	13469
	MS Silverlight	938	1813	3545	5545	7195	9412	23645
	Oracle JavaFX	1888	3888	6392	9476	11471	15584	-
Google Chrome 4.0	Adobe Flex	533	1617	2611	3484	4826	5898	13231
	MS Silverlight	763	1778	3150	4805	6602	8532	19986
	Oracle JavaFX	2043	4115	6350	10556	11948	15624	-

Tab. 9.1 Výsledky testování rychlosti načítání grafických prvků (vyjádřeno v milisekundách)

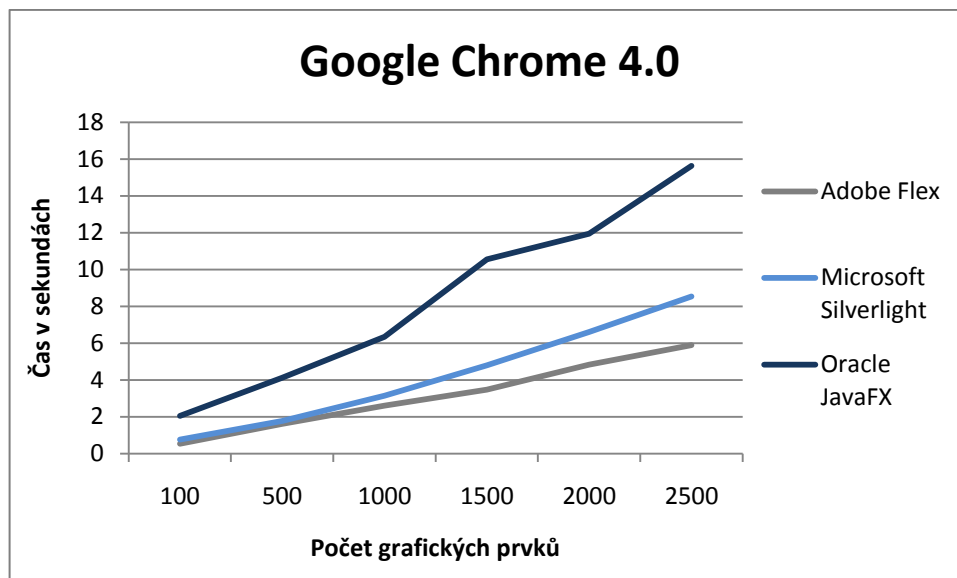
V grafech 9.1, 9.2 a 9.3 lze vidět závislost doby načítání testovacích aplikací na počtu grafických prvků podle webového prohlížeče. V naměřených hodnotách technologií Adobe Flex a Microsoft Silverlight se nevyskytují žádné odchylky a křivky na grafu jsou téměř lineární. Rozdíl je ovšem u technologie JavaFX, kde docházelo k mírným odchylkám



Graf 9.1 Závislost doby načítání aplikace na počtu grafických prvků (Internet Explorer 8.0)



Graf 9.2 Závislost doby načítání aplikace na počtu grafických prvků (Mozilla Firefox 3.6.2)



Graf 9.3 Závislost doby načítání aplikace na počtu grafických prvků (Google Chrome 4.0)

## 9.2 Test rychlosti zpracování složitějšího algoritmu

V tomto testu jsem porovnával rychlost vybraných technologií při zpracování složitého výpočtu. Za vhodný příklad jsem zvolil algoritmus pro nalezení prvočísel. Vytvořené aplikace zjišťují počet prvočísel menších než je zadaná hodnota. Odborně se tento algoritmus pro nalezení prvočísel nazývá Eratosthenovo síto. Časová složitost tohoto algoritmu je  $O(N \cdot \log(\log N))$ , kde  $N$  je horní mez rozsahu.

Aby měl test co největší vypovídající hodnotu, zvolil jsem pro výpočet velký rozsah hodnot. Nejmenší zadaná hodnota pro výpočet je 10 000, naopak nejvyšší hodnota je 10 miliónů.

### 9.2.1 Způsob měření

Měření probíhalo podobným způsobem jako u předcházejícího testu. Před spuštěním procesu byl zaznamenán aktuální čas, který byl porovnán s časem při dokončení celého výpočtu. Následně po vypočítání je v aplikaci zobrazen výsledek a čas zpracování.

### 9.2.2 Výsledky testování

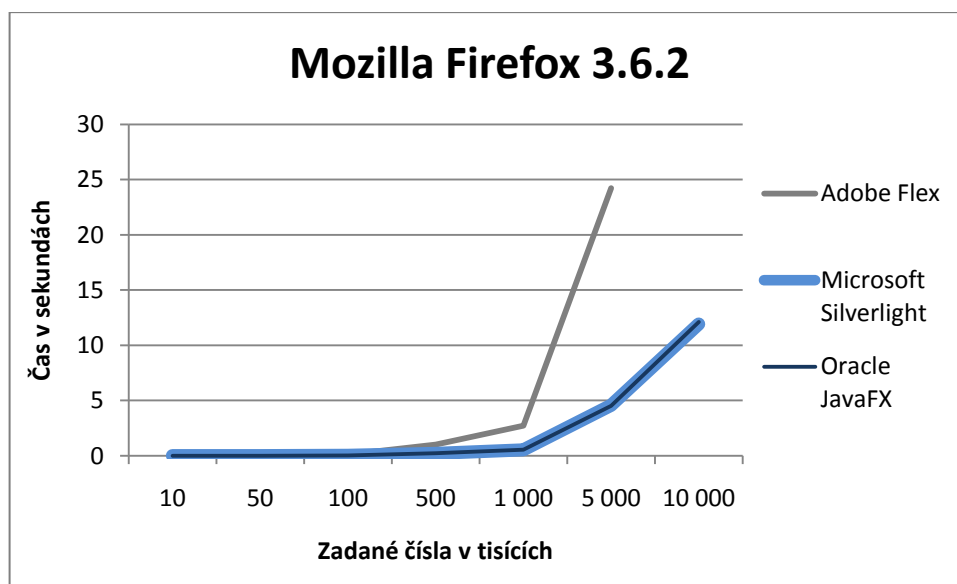
Z výsledků, které jsou zobrazeny v tabulce 9.2, můžeme vidět, že nejlépe si při výpočtech počínala technologie Silverlight. Jen o nepatrně vyšší čas výpočtu dosáhla technologie JavaFX, v ojedinělých případech dosáhla lepších výsledků než Silverlight. Nejhorší v tomto testu dopadla technologie Adobe Flex. Časové rozdíly lze vidět v tabulce již při menších hodnotách, dokonce při výpočtu se zadanou hodnotou 10 miliónů si neporadila vůbec.

Z pohledu výkonnosti pluginů v jednotlivých prohlížečích dopadly výsledky velmi podobně. Nelze tedy přímo určit, který prohlížeč byl při práci s pluginy výkonnostně nejlepší, případně nejhorší.

		Zadané čísla pro výpočet						
		10 000	50 000	100 000	500 000	1 000 000	5 000 000	10 000 000
Internet Explorer 8.0	Adobe Flex	6	65	132	1023	2644	24714	-
	MS Silverlight	5	18	37	217	534	4440	11527
	JavaFX	5	19	46	239	549	4549	11931
Mozilla Firefox 3.6.2	Adobe Flex	12	63	125	1006	2714	24233	-
	MS Silverlight	5	17	48	222	542	4552	11916
	JavaFX	6	13	32	226	561	4506	12102
Google Chrome 4.0	Adobe Flex	5	76	148	1005	2561	24596	-
	MS Silverlight	5	18	38	226	528	4540	11686
	JavaFX	7	19	34	229	579	4594	11643

Tab. 9.2 Výsledky testu rychlosti zpracování algoritmu pro nalezení prvočísel (vyjádřeno v milisekundách)

Výsledné hodnoty jsou také promítnuty v grafu 9.4. Lze si v něm všimnout velkého časového skoku výpočtu technologie Flex mezi zadanými hodnotami 1 a 5 miliónů. Je uveden pouze graf pro webový prohlížeč Mozilla Firefox, jelikož grafy pro ostatní prohlížeče byly obdobné. Hodnoty Silverlight a JavaFX jsou téměř totožné a proto se v grafu překrývají.



Graf 9.4 Závislost doby zpracování výpočtu na velikosti zadané hodnoty (Mozilla Firefox 3.6.2)

## 10 Závěr

V současné době se Internet stal nedílnou součástí života většiny populace. Nejen mladá generace tráví svůj volný čas na Internetu a využívá nepřeberné množství jeho informací. Neustále se proto vyvíjejí technologie, které usnadňují a rozšiřují možnosti poskytování služeb uživatelům. Část těchto technologií se používá pro vytváření takzvaných RIA aplikací.

Tato práce se zabývá možnostmi RIA aplikací. V teoretické části byl vysvětlen význam termínu RIA a popsány výhody a nevýhody těchto aplikací. V dalších částech práce jsou popsány existující technologie používané pro vývoj interaktivních RIA aplikací.

Praktická část této diplomové práce zahrnovala porovnání zvolených technologií. Zvolil jsem si tři, dle mého uvážení nejpoužívanější technologie: Adobe Flex 3, Microsoft Silverlight 3 a Oracle JavaFX 1.2. Každé z těchto technologií jsem věnoval samostatnou kapitolu, ve které jsem popsal jejich historii, architekturu, aktuální verzi, možnosti vývojových nástrojů, běhové prostředí a jejich nasazení. Následně jsem vybrané technologie porovnal z hlediska jejich základních vlastností, efektivity programování a jejich výkonů.

Vítězem mého porovnávání bych označil technologii Adobe Flex 3. Důvodů, proč tato technologie předčila své konkurenty, je celá řada. Hlavní předností této technologie je běhové prostředí Adobe Flash Player. Tento přehrávač je multiplatformní a je rozšířen na většině počítačů bez ohledu na operační systém. Dalším důvodem je snadno naučitelný jazyk ActionScript a jednoduchý vývoj aplikací. Tato technologie dosáhla také nejlepších výsledků v testu zaměřeném na rychlost načítání a vykreslování grafických prvků. Nevýhodami, které bych vytknul, jsou výsledná velikost souboru a možnosti vytváření animací. Jediný vývojový nástroj Adobe Builder je placený. Adobe Flex není určen pro náročné výpočty, jak bylo ukázáno v testu rychlosti zpracování složitějšího algoritmu.

Na druhém místě, dle mého názoru, dopadla technologie Microsoft Silverlight 3. Jedná se o mladší technologii než je Adobe Flex, ale začíná této technologii pomalu konkurovat. Hlavní výhodou technologie Silverlight je .NET Framework, který podporuje dva programovací jazyky (C# a VB.NET) a nabízí rozsáhlejší knihovny než technologie Flex. Pro tuto technologii také existují bohatší vývojová prostředí, umožňující snadnější vytváření aplikací. Nutno dodat, že některá z vývojových prostředí jsou zdarma. Výhodou je snadné vytváření animací, postaveného na time-based modelu. V testu zabývajícím se zpracováním algoritmu dopadl Silverlight nejlépe a doporučil bych ho na vytváření složitějších webových aplikací. Za hlavní nevýhodu této technologie bych označil rozšířenost běhového prostředí, které oproti ostatním zvoleným technologiím je velice malé.

Nejmladší technologii pro vývoj RIA aplikací je Oracle JavaFX 1.2. Tato technologie je stále na začátku svého vývoje a osobně ji ještě nepovažuji za konkurenceschopnou vůči ostatním vybraným technologiím. Nespornou předností této technologie jsou nulové náklady na vývojové prostředí - všechny jsou zdarma k dispozici. Výhodou je také velikost výsledného souboru, která je velice malá. V testu zaměřeném na rychlost zpracování složitějšího algoritmu dosahoval JavaFX obdobných výsledků jako Silverlight. Na této technologii jsem našel více nedostatků než samotných výhod. Za hlavní nedostatek považuji rychlost spouštění a zobrazení aplikace; v daném testu dopadla tato technologie nejhůře. Dostupné vývojové prostředí nedosahuje takových



kvalit jako u konkurence a nástrojů pro usnadnění programování je málo. Samotný jazyk JavaFX Script mi připadá komplikovaný a programování v tomto jazyce je velice zdlouhavé.

Ačkoliv jsem zvolil za vítěze technologii Adobe Flex 3, všechny vybrané technologie jsou stále ve vývoji. Nutno dodat, že technologií pro vývoj RIA aplikací je více a konkurence je vysoká. Osobně si myslím, že boj o dominanci na trhu v oblasti RIA bude ještě dlouho probíhat. Nové verze vybraných technologií by měly vyjít během letošního roku a můžeme se těšit na mnoho dalších rozšíření a zlepšení.

Tuto práci lze v budoucnu rozšířit o porovnání s dalšími technologiemi pro vývoj RIA aplikací. Případně by bylo možné na práci navázat a porovnat novější verze vybraných technologií.

## Seznam použité literatury

### Internetové zdroje

- [1] Rich Internet application In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 7. 11. 2004, 13. 6. 2009 [cit. 23. 6. 2009]. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Rich\\_Internet\\_application](http://en.wikipedia.org/wiki/Rich_Internet_application)>.
- [2] BERNARD, Borek. *Interval.cz* [online]. 25. 4. 2008 [cit. 23. 6. 2009]. Rich Internet Applications v roce 2008. Dostupné z WWW: <<http://interval.cz/clanky/rich-internet-applications-v-roce-2008>>.
- [3] PICHLÍK, Roman. *Interval.cz* [online]. 14. 6. 2005 [cit. 23. 6. 2009]. Rich Internet Application. Dostupné z WWW: <<http://interval.cz/clanky/rich-internet-application>>.
- [4] DAVIS, Alek. *Docstoc.com* [online]. 2008 [cit. 23. 6. 2009]. Google I/O 2008: Rich Internet Application (RIA) Development . Dostupné z WWW: <<http://www.docstoc.com/docs/5170530/Google-IO-2008-Rich-Internet-Application-%28RIA%29-Development>>.
- [5] MORAVEC, Zdeněk. *Programujte.com* [online]. 14. 4. 2009 [cit. 23. 6. 2009]. RIA - Rich Internet Applications. Dostupné z WWW: <<http://programujte.com/?akce=clanek&cl=2009041200-ria-rich-internet-applications>>.
- [6] *Canoo.com* [online]. 16. 8. 2007 [cit. 23. 6. 2009]. Adding AJAX to existing Web Apps. Dostupné z WWW: <[canoo.com/blog/2007/08/16/ajax-unforeseen-and-expensive-refactorings-in-lower-layers-part-2-of-3](http://canoo.com/blog/2007/08/16/ajax-unforeseen-and-expensive-refactorings-in-lower-layers-part-2-of-3)>.
- [7] MATULÍK, Petr. *Morosystems.cz* [online]. 28. 6. 2009 [cit. 23. 5. 2005]. AJAX – web či desktop?. Dostupné z WWW: < <http://weblog.morosystems.cz/ostatni/ajax-web-ci-desktopapplication>>.
- [8] AJAX In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 11. 10. 2005, 16. 4. 2009 [cit. 28. 6. 2009]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/AJAX>>.
- [9] Adobe Flash In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 6. 1. 2006, 21. 6. 2009 [cit. 30. 6. 2009]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Adobe\\_Flash](http://cs.wikipedia.org/wiki/Adobe_Flash)>.
- [10] Curl (programming language) In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 2. 3. 2002, 9. 6. 2009 [cit. 30. 6. 2009]. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Curl\\_%28programming\\_language%29](http://en.wikipedia.org/wiki/Curl_%28programming_language%29)>.
- [11] *Flickr.com* [online]. 2009 [cit. 30. 6. 2009]. Curl Technology - How it Works. Dostupné z WWW: <<http://www.flickr.com/photos/curltech/449446475>>.
- [12] VESELKA, Aleš. *Symbio.cz* [online]. 5. 2. 2007 [cit. 30. 6. 2009]. Obohaťte své uživatele pomocí RIA. Dostupné z WWW: <<http://www.symbio.cz/clanky/obohatte-sve-uzivatele-pomoci-ria.html>>.
- [13] *Lupa.cz* [online]. 31. 5. 2007 [cit. 30. 6. 2009]. Google Gears - online aplikace offline. Dostupné z WWW: <<http://jilm.blog.lupa.cz/2007/05/31/google-gears-online-aplikace-offline>>.

- [14] Gears (software) In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 31. 5. 2007, 22. 11. 2009 [cit. 28. 11. 2009]. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Gears\\_%28software%29](http://en.wikipedia.org/wiki/Gears_%28software%29)>.
- [15] *Code.google.com* [online]. 2009 [cit. 28. 11. 2009]. Gears API. Dostupné z WWW: <<http://code.google.com/intl/cs/apis/gears/architecture.html>>.
- [16] ČÍŽEK, Jakub. *Zive.cz* [online]. 5. 2. 2008 [cit. 29. 11. 2009]. Mozilla Prism: Na web přes ikonku. Dostupné z WWW: <<http://www.zive.cz/Clanky/Mozilla-Prism-Na-web-pres-ikonku/sc-3-a-140180/default.aspx>>.
- [17] Prism In *MozillaWiki* [online]. Mozilla Foundation, 15. 7. 2007, 16. 7. 2009 [cit. 29. 11. 2009]. Dostupné z WWW: <<https://wiki.mozilla.org/Prism>>.
- [18] *Mozillalabs.com* [online]. 24. 10. 2007 [cit. 29. 11. 2009]. Introducing Prism. Dostupné z WWW: <<https://mozillalabs.com/blog/2007/10/prism>>.
- [19] Adobe Flex In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 22. 11. 2004, 9. 12. 2009 [cit. 12. 12. 2009]. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Adobe\\_Flex](http://en.wikipedia.org/wiki/Adobe_Flex)>.
- [20] RAZA, Imam. *Imamrazaansari.blogspot.com* [online]. 17. 11. 2008 [cit. 12. 12. 2009]. Adobe Flex. Dostupné z WWW: <<http://imamrazaansari.blogspot.com/2008/11/adobe-flex.html>>.
- [21] *Amsoft.cz* [online]. 2008 [cit. 12. 12. 2009]. Adobe Flex 3. Dostupné z WWW: <<http://www.amsoft.cz/produkty/Adobe/flex/overview.html>>.
- [22] MXML In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 12. 8. 2004, 7. 12. 2009 [cit. 14. 12. 2009]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/MXML>>.
- [23] ActionScript In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 19. 1. 2006, 1. 6. 2009 [cit. 14. 12. 2009]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/ActionScript>>.
- [24] *Flash.jakpsatweb.cz* [online]. 2010 [cit. 14. 12. 2009]. Úvod do ActionScriptu 3.0. Dostupné z WWW: <<http://flash.jakpsatweb.cz/actionscript-3>>.
- [25] *Amsoft.cz* [online]. 2008 [cit. 14. 12. 2009]. Adobe Flex 3. Dostupné z WWW: <<http://www.amsoft.cz/produkty/Adobe/flex/builder.html>>.
- [26] BERNARD, Borek. *Zdrojak.root.cz* [online]. 21. 10. 2009 [cit. 15. 12. 2009]. Novinky pro Flash/Flex vývojáře. Dostupné z WWW: <<http://zdrojak.root.cz/clanky/novinky-pro-flash-flex-vyvojare>>.
- [27] BERNARD, Borek. *Interval.cz* [online]. 4. 6. 2008 [cit. 15. 12. 2009]. Adobe Flex - co je a co není. Dostupné z WWW: <<http://interval.cz/clanky/adobe-flex-co-je-a-co-neni>>.
- [28] Microsoft Silverlight In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 26. 2. 2007, 15. 1. 2010 [cit. 15. 1. 2010]. Dostupné z WWW: <[http://en.wikipedia.org/wiki/Microsoft\\_Silverlight](http://en.wikipedia.org/wiki/Microsoft_Silverlight)>.
- [29] LACKO, Ľuboslav. *Blog.aspnet.sk* [online]. 3. 11. 2009 [cit. 15. 1. 2010]. Silverlight 3.0: Stručný pohľad do histórie platformy Silverlight. Dostupné z WWW: <<http://blog.aspnet.sk/knihy/archive/2009/11/03/strucny-pohlad-do-historie-platformy-silverlight.aspx>>.

- [30] JELÍNEK, Jan. *Zdrojak.root.cz* [online]. 30. 3. 2009 [cit. 18. 1. 2010]. Představení Silverlightu 3 beta 1. Dostupné z WWW: <<http://zdrojak.root.cz/clanky/predstaveni-silverlightu-3-beta-1>>.
- [31] Xaml In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 24. 7. 2007, 24. 7. 2007 [cit. 18. 1. 2010]. Dostupné z WWW: <<http://en.wikipedia.org/wiki/Xaml>>.
- [32] HERCEG, Tomáš. *Vbnet.cz* [online]. 18. 4. 2009 [cit. 18. 1. 2010]. Základní elementy VB.NET a C#. Dostupné z WWW: <[http://www.vbnet.cz/clanek--126-zakladni-elementy\\_vb\\_net\\_a\\_c\\_.aspx](http://www.vbnet.cz/clanek--126-zakladni-elementy_vb_net_a_c_.aspx)>
- [33] BECHYNSKÝ, Štěpán. *Zdrojak.root.cz* [online]. 24. 7. 2009 [cit. 18. 1. 2010]. Silverlight 3: ukázky novinek pro vývojáře. Dostupné z WWW: <<http://zdrojak.root.cz/clanky/silverlight-3-ukazky-novinek-pro-vyvojare>>.
- [34] *Eclipse4sl.org* [online]. 2006 [cit. 18. 1. 2010]. Eclipse Tools for Microsoft Silverlight. Dostupné z WWW: <<http://www.eclipse4sl.org>>.
- [35] *Microsoft.com* [online]. 1. 12. 2009 [cit. 20. 1. 2010]. Vyzkoušejte beta verzi Silverlight 4. Dostupné z WWW: <[http://www.microsoft.com/cze/presspass/msg/20091201\\_news1.msp](http://www.microsoft.com/cze/presspass/msg/20091201_news1.msp)>.
- [36] *Virklis.cz* [online]. 2008 [cit. 24. 1. 2010]. Sun Microsystems uvádí platformu JavaFX 1.0 s bohatými multimediálními funkcemi pro 800 milionů počítačů s Javou na trhu. Dostupné z WWW: <<http://www.virklis.cz/cs/press/sun-microsystems/javafx1.html>>.
- [37] DOEDERLEIN, Osvaldo Pinali. *Weblogs.java.net* [online]. 13. 2. 2009 [cit. 24. 1. 2010]. JavaFX 1.1 released, some first impressions. Dostupné z WWW: <[http://weblogs.java.net/blog/opinali/archive/2009/02/javafx\\_11\\_relea.html](http://weblogs.java.net/blog/opinali/archive/2009/02/javafx_11_relea.html)>.
- [38] *Javafx.com* [online]. 2009 [cit. 10. 2. 2010]. Getting Started With JavaFX Production Suite. Dostupné z WWW: <[http://javaafx.com/docs/gettingstarted/production\\_suite](http://javaafx.com/docs/gettingstarted/production_suite)>.
- [39] CARLE, Jim. *Informit.com* [online]. 2. 6. 2009 [cit. 10. 2. 2010]. What's New in JavaFX 1.2?. Dostupné z WWW: <<http://www.informit.com/articles/article.aspx?p=1352545>>.
- [40] *Steveonjava.com* [online]. 31. 5. 2009 [cit. 10. 2. 2010]. JavaFX 1.2 Top 10 and Migration Guide. Dostupné z WWW: <<http://steveonjava.com/2009/05/31/javafx-1-2-top-10>>.
- [41] *Blogs.sun.com* [online]. 4. 12. 2008 [cit. 10. 2. 2010]. Remember Write Once, Run Anywhere?. Dostupné z WWW: <[http://blogs.sun.com/amy/entry/remember\\_write\\_once\\_run\\_anywhere](http://blogs.sun.com/amy/entry/remember_write_once_run_anywhere)>.
- [42] JavaFX In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 4. 6. 2009, 23. 9. 2009 [cit. 12. 2. 2010]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/JavaFX>>.
- [43] LEONARD, Anghel. *Introduction to JavaFX Script* [online]. 8. 1. 2007 [cit. 12. 2. 2010]. Onjava.com. Dostupné z WWW: <<http://onjava.com/pub/a/onjava/2007/07/27/introduction-to-javafx-script.html>>.
- [44] Java (programovací jazyk) In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida): Wikipedia Foundation, 22. 6. 2004, 21. 11. 2009 [cit. 12. 2. 2010]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Java\\_%28programovac%C3%AD\\_jazyk%29](http://cs.wikipedia.org/wiki/Java_%28programovac%C3%AD_jazyk%29)>.
- [45] *Javafx.com* [online]. 2010 [cit. 15. 2. 2010]. JavaFX: General Questions. Dostupné z WWW: <<http://javaafx.com/faq>>.

- [46] ČERNÝ, Ondřej. *Vse.cz* [online]. 2006 [cit. 15. 2. 2010]. Co je to Java web start - základní popis. Dostupné z WWW: <<http://st.vse.cz/~XCERO08>>.
- [47] WEAVER, James. *Learnjavafx.typepad.com* [online]. 3. 12. 2009 [cit. 17. 2. 2010]. JavaFX 1.3 Leakage at Devvxx 2009. Dostupné z WWW: <<http://learnjavafx.typepad.com/weblog/2009/12/javafx-13-leakage-at-devvxx-and-%C3%B8redev.html>>.
- [48] *Sellmic.com* [online]. 5. 6. 2009 [cit. 17. 2. 2010]. JavaFX Authoring tool demo at JavaOne 2009 (with video) – updated. Dostupné z WWW: <<http://sellmic.com/blog/2009/06/05/javafx-authoring-tool-demo-at-javaone-2009-with-video>>.
- [49] *Riastats.com* [online]. 2009 [cit. 17. 2. 2010]. Rich Internet Application Statistics. Dostupné z WWW: <<http://www.riastats.com>>.
- [50] *Javafx.com* [online]. 2010 [cit. 23. 3. 2010]. Creating a Media Player. Dostupné z WWW: <<http://javafx.com/docs/articles/media/player.jsp>>.

## Seznam obrázků

Obr. 2.1 Oblasti RIA aplikací .....	5
Obr. 2.2 Model komunikace klasické webové aplikace.....	7
Obr. 2.3 Obecný model komunikace RIA aplikací .....	7
Obr. 3.1 Architektura webových aplikací .....	9
Obr. 3.2 Model komunikace AJAX aplikace .....	10
Obr. 3.3 Model Curl aplikace .....	11
Obr. 3.4 Model OpenLaszlo aplikace .....	12
Obr. 3.5 Návrh architektury aplikace od firmy Google umožňující online/offline práci .....	13
Obr. 3.6 Mozilla Prism .....	14
Obr. 4.1 Sada nástrojů pro Adobe Flex 2.0 .....	16
Obr. 4.2 Framework Flex 3 .....	17
Obr. 4.3 Struktura Flex aplikace .....	19
Obr. 5.1 Architektura Microsoft Silverlight 3.0 .....	22
Obr. 5.2 Struktura Silverlight aplikace .....	26
Obr. 6.1 Platforma JavaFX .....	29
Obr. 6.2 Struktura JavaFX aplikace .....	33
Obr. 8.1 Grafický návrh přehrávače.....	39
Obr. 8.2 JavaFX Media API .....	42

## Seznam tabulek

Tab. 2.1 Nejznámější technologie pro tvorbu RIA aplikací v jednotlivých oblastech.....	5
Tab. 7.1 Souhrn základních vlastností vybraných technologií.....	35
Tab. 7.2 Velikost výsledných souborů ukázkového příkladu "HelloWorld" .....	36
Tab. 7.3 Souhrn vlastností vývojových prostředí pro dané technologie .....	37
Tab. 7.4 Rozšířenost běhového prostředí v závislosti na webovém prohlížeči (v procentuálním vyjádření) .....	38
Tab. 7.5 Rozšířenost běhového prostředí v závislosti na operačním systému (v procentuálním vyjádření) .....	38
Tab. 8.1 Podpora formátu ve vybraných technologiích .....	40
Tab. 9.1 Výsledky testování rychlosti načítání grafických prvků (vyjádřeno v milisekundách) .....	45
Tab. 9.2 Výsledky testu rychlosti zpracování algoritmu pro nalezení prvočísel (vyjádřeno v milisekundách) .....	48

## Seznam výpisů

Výpis 4.1 Základní deklarace MXML .....	17
Výpis 4.2 Vytvoření tlačítka v jazyce MXML.....	18
Výpis 4.3 Vytvoření tlačítka v jazyce ActionScript 3.0.....	18
Výpis 4.4 Ukázková Flex aplikace Hello World.....	18
Výpis 4.5 HTML kód Flex aplikace vkládaný do webové stránky (ukázka Hello World).....	20
Výpis 5.1 Základní deklarace XAML .....	23
Výpis 5.2 Vytvoření tlačítka v jazyce XAML .....	23
Výpis 5.3 Vytvoření tlačítka v jazyce C# .....	23
Výpis 5.4 Vytvoření tlačítka v jazyce Visual Basic .NET .....	23
Výpis 5.5 Ukázková Silverlight aplikace Hello World (vzhled aplikace - XAML) .....	24
Výpis 5.6 Ukázková Silverlight aplikace Hello World (funkcionalita aplikace - C#) .....	25
Výpis 5.7 Silverlight aplikace Hello World (funkcionalita aplikace - VB.NET) .....	25
Výpis 5.8 HTML kód Silverlight aplikace vkládaný do webové stránky (ukázka Hello World) ....	27
Výpis 6.1 Základní deklarace JavaFX .....	30
Výpis 6.2 Vytvoření tlačítka v jazyce JavaFX (přetažení z palety) .....	30
Výpis 6.3 Nadefinování tlačítka v jazyce JavaFX .....	31
Výpis 6.4 Vytvoření tlačítka v jazyce Java .....	31
Výpis 6.5 Ukázková JavaFX aplikace Hello World .....	32
Výpis 6.6 HTML kód JavaFX aplikace vkládaný do webové stránky (ukázka Hello World).....	33
Výpis 6.7 Kód souboru HelloWorld_browser.jnlp .....	34
Výpis 8.1 Komponenta „videoDisplay“ ve vytvořené aplikaci VideoPlayer (Adobe Flash).....	41
Výpis 8.2 Komponenta „MediaPlayer“ ve vytvořené aplikaci VideoPlayer (Microsoft Silverlight) .....	41
Výpis 8.3 „MediaPlayer“ a „MediaView“ ve vytvořené aplikaci VideoPlayer (Oracle JavaFX) ...	42

## Seznam grafů

Graf 9.1 Závislost doby načítání aplikace na počtu grafických prvků (Internet Explorer 8.0) .....	46
Graf 9.2 Závislost doby načítání aplikace na počtu grafických prvků (Mozilla Firefox 3.6.2) .....	46
Graf 9.3 Závislost doby načítání aplikace na počtu grafických prvků (Google Chrome 4.0).....	47
Graf 9.4 Závislost doby zpracování výpočtu na velikosti zadané hodnoty (Mozilla Firefox 3.6.2) .....	48

## Seznam příloh

Příloha č. 1      CD

- |   |                                 |
|---|---------------------------------|
| • tento dokument                            | kořenový adresář                |
| • zdrojové kódy aplikací                    | ./Zdrojové kódy                 |
| • aplikace HelloWorld                       | ./Zdrojové kódy/HelloWorld      |
| • vytvořené videopřehrávače                 | ./Zdrojové kódy/VideoPlayer     |
| • testovací aplikace pro vykreslování prvků | ./Zdrojové kódy/RenderingTest   |
| • testovací aplikace pro výpočet            | ./Zdrojové kódy/ComputationTest |
| • vlastní obrázky použité v dokumentu       | ./Obrázky                       |